Section IV:

AMENDMENT UNDER 37 CFR §1.121

REMARKS

Rejections under 35 U.S.C. §112

In the Office Action, the examiner has rejected claim 1 under 35 U.S.C. §112, second paragraph, for reasons of lacking antecedent basis for "said problem management" and "failed message server". Examiner proposed a correction, which was assumed for the remaining rejections.

Examiner's proposed correction and assumed meaning of this claim was correct, and the present amendment formalizes adoption of examiner's suggestion. The same change also applies to Claim 12.

Rejections under 35 U.S.C. §103

In the Office Action, examiner has rejected claims 1 - 5, and 9 - 10 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6, 061,723 to Walker, et al. (hereinafter "Walker"). Examiner has stated that Walker teaches all of the claimed steps or elements as set forth in our independent claims 1 and 12, except for a difference in setting a predetermined timer in our disclosed manner. Examiner has not stated where motivation to modify Walker's timer is found in the Walker disclosure, or if examiner considers this to be of his or her own knowledge.

We respectfully traverse examiner's holding that Walker teaches using or executing a "fault tree analysis" as we have done. Walker teaches a "bottoms up" process:

> A seventh advantage is an ability to handle arbitrary topologies. Algorithms that attempt to find a root cause of a network element are easily fooled because of the complexity in customer network configurations (e.g., they may contain loops and dynamic routing). Algorithms in this implementation are **"Bottoms up"** and will not be fooled into thinking a network element is down when a redundant router fails. (Col. 4, lines 52 - 58, emphasis added)

Walker also teaches evaluation of events with respect to their priority or order of issuance/receipt:

> A last advantage is **"event ordering."** During network failure situations, confusion is compounded by implementations that discover inaccessible network elements in random order. **This implementation contains new queuing algorithms to discover failures in a predictable order.** This predictability is helpful to both the network administrator, and other event correlation processes that a user or third parties might construct. (Col. 4, line 64 to Col. 5, line 4, emphasis added)

Third, Walker teaches use of signal paths, or "criticalRoute attributes", in their failure analysis:

> Network management apparatus for distinguishing between broken and inaccessible network elements in a network, and for presenting this information to a network administrator in an easy to comprehend format, is shown in FIGS. 5, 7 & 8. The apparatus generally comprises a display process 104, 120 and a network monitor 110, connected by way of one or more event buses 114. The network monitor 110 comprises a means for discovering the topology of a plurality of network elements 124, 128-136 connected thereto, means for periodically polling a plurality of network interfaces associated with the plurality of network elements 124, 128-136, means for computing or validating a **criticalRoute attribute** for each of the plurality of network interfaces, and means for analyzing the status of network interfaces (e.g., N.1, A.1, A.2, B.1, B.2, C.1, C.2, X.1, Y.1, Z.1) identified by the **criticalRoute attribute** of an interface in question (IIQ) which is not responding to a poll.
>
> Likewise, a computer implemented method of distinguishing between broken and inaccessible network elements, and for presenting this information to a network administrator in an easy to comprehend format, may comprise the steps of 1) discovering the topology of a

> plurality of network elements 124, 128-136, 2) periodically polling a plurality of network interfaces associated with the plurality of network elements 124, 128-136, 3) computing or validating a **criticalRoute attribute** for each of the plurality of network interfaces, and 4) analyzing the status of network interfaces identified by the **criticalRoute attribute** of an interface in question (IIQ) which is not responding to a poll. (Col. 5, lines 35 - 63, emphasis added)

This type of order-dependent, bottoms-up event-drive analysis appears to be similar to the well known "Event Tree Analysis", and appears to be similar to "path analysis" based upon Walker's used of critical path attributes (e.g. route attributes).

However, we have claimed in all of our claims the used of "Fault Tree Analysis", which is well-known to be a "tops down" approach employing logical rules or Boolean equations, which is not an event-driven process and is not subject to ordering of reported events. A Fault Tree Analysis also generates a likelihood for each element as being the failed element (as we have claimed), which is not available from Event Tree Analysis. It is well known in the art that Fault Tree Analysis and Event Tree Analysis are not similar or equivalent.

In Appendix A to this reply, Pat L. Clemens and Jacobs Sverdup (hereinafter "Clemens) have documented on www.fault-tree.net the general characteristics of Event Tree Analysis in comparison to Fault Tree Analysis, and including the advantages and disadvantages of each. Especially note the following points made by Clemens:

(a) that Event Tree Analysis is a "bottoms up" analysis (Clemens page 2);

(b) Event Tree Analysis is event sequence or order dependent (Clemens page 12 "sequence dependent scenarios are not modeled well");

(c) Event Tree Analysis is *complementary to* but not the same as Fault Tree Analysis (Clemens page 2);

(d) Event Tree Analysis is path oriented (Clemens page 4)

(e) Event Tree Analysis does not determine probability of each failure possibility (Clemens page 5 "fault tree analysis ... may be necessary to determine the probability...);

(f) and that Event Trees are not the same as Fault Trees (Clemens illustration on page 10 wherein Fault Trees include logical operators such as OR gates).

Our assertions of the characteristics of Fault Tree Analysis are supported by the publicly available paper "Fault Tree Analysis - A History" by Clifton A. Ericson II (hereinafter "Ericson"), reproduced in Appendix B of the present reply, and especially that:

(a) Fault Tree Analysis employs a "top down" technique (Ericson page 2, right column, under "Highlights"; and

(b) Fault Tree Analysis uses a logical model, Boolean algebra and probability (Ericson page 1 under "Fundamental Concept of FTA").

For further illustration that these definitions of Fault Tree Analysis and Event Tree Analysis are conventional and well known, we also reproduce "Fault Tree Analysis Methods and Applications" by James M. Kelly in our Appendix C, "What is Fault Tree Analysis?" by Relex Software, and "Safety Critical Systems Analysis" by Robert Slater.

These documents are presented as evidence of the well-agreed upon definitions of these terms, and do not necessarily represent prior art to our invention, as our invention claims our uniquecombination of Fault Tree Analysis with several other elements and steps relating specifically to diagnosis of network faults and reduction of fault message traffic.

Also in the Office Action, examiner has rejected claims 7 - 8, 11 - 13, 15 - 23, and 25 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6, 061,723 to Walker, et al. (hereinafter "Walker") in view of U.S. Patent No. 6,564,341 to Sundaram, *et al.* (hereinafter "Sundaram"). Each of these claims, however, claims the use of Fault Tree Analysis, for which Sundaram is silent.

Also in the Office Action, examiner has rejected claims 24 and 26 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6, 061,723 to Walker, et al. (hereinafter "Walker") in view of U.S. Patent No. 6,564,341 to Sundaram, *et al.* (hereinafter "Sundaram") in further view of U.S. Patent No. 6,571,285 to Groath, *et al.* (hereinafter "Groath"). Claims 24 and 26, however, also claim Fault Tree Analysis as a step, element or limitation, for which Groath is silent.

We therefore submit that our claims are patentable for the following reasons:

1. The combination or modification of the references in the manner suggested by the examiner would render the primary reference inoperable or unsatisfactory for its intended purpose. MPEP § 2143.01 states:

> If [the] proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification.

The facts derived from the references and set forth below indicated that the suggested combination or modification would render the primary reference inoperable or unsatisfactory for its intended purpose. Therefore, the rejection is unsupported by the cited art, and its withdrawal is requested.

i. Walker teaches that their event tree type of analysis is needed or required to avoid confusion of the diagnostics. (Walker Col. 4, lines 52 - 58)

2. The combination or modification of the references in the manner suggested by the examiner would change the principle of operation of the primary reference. MPEP §2143.01 states:

> If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.

The facts derived from the references and set forth below indicated that the examiner's suggested combination or modification would change the principle of operation of the primary reference. Therefore, the rejection is unsupported by the cited art, and its withdrawal is requested.

i. The primary reference employs a bottoms-up, event driven, order dependent analysis instead of a top-down, logic model.

3. The reference(s) teach away from the examiner's proposed combination. MPEP §2145 states:

> It is improper to combine references where the references teach away from their combination.

The facts derived from the references and set forth below indicated that the references teach away from their combination. Therefore, the rejection is unsupported by the cited art, and its withdrawal is requested.

i. It would be improper to combine Walker with a reference which discloses Fault Tree Analysis as this would be contrary to Walker's assertion that a bottoms-up technique is needed to avoid the system being "fooled". (Walker Col. 4, lines 52 - 58)

4. The combination or modification of the references in the manner suggested by the examiner does not teach all the claimed elements, steps, or restrictions. MPEP §2143.03 states:

> **All Claim Limitations Must Be Taught or Suggested.** To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art.

The facts derived from the references and set forth in the preceding remarks indicate that the examiner's suggested combination and modification of the cited references does not teach all claimed elements, limitations or steps, namely use of Fault Tree Analysis. Therefore, the rejection is unsupported by the art and should be withdrawn.

Conclusion

Reconsideration of all rejections is hereby requested in view of the present amendment and remarks.

Appendix A

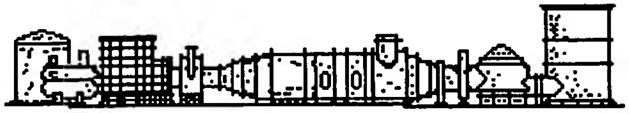
EVENT TREE ANALYSIS

2nd Edition

P. L. Clemens

June 1990

Sverdrup



provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

EVENT TREE ANALYSIS IS...

- A bottom-up, deductive, system safety analytical technique
- Applicable to:
    - Physical systems, with or without human operators
    - Decision-making / management systems
- Complementary to other techniques, e.g....
    - Fault Tree Analysis
    - Failure Modes and Effects Analysis

EVENT TREE ANALYSIS...

Explores system **RESPONSES**

to

Initiating **“CHALLENGES”**

and

Enables **PROBABILITY ASSESSMENT**

of

SUCCESS / FAILURE

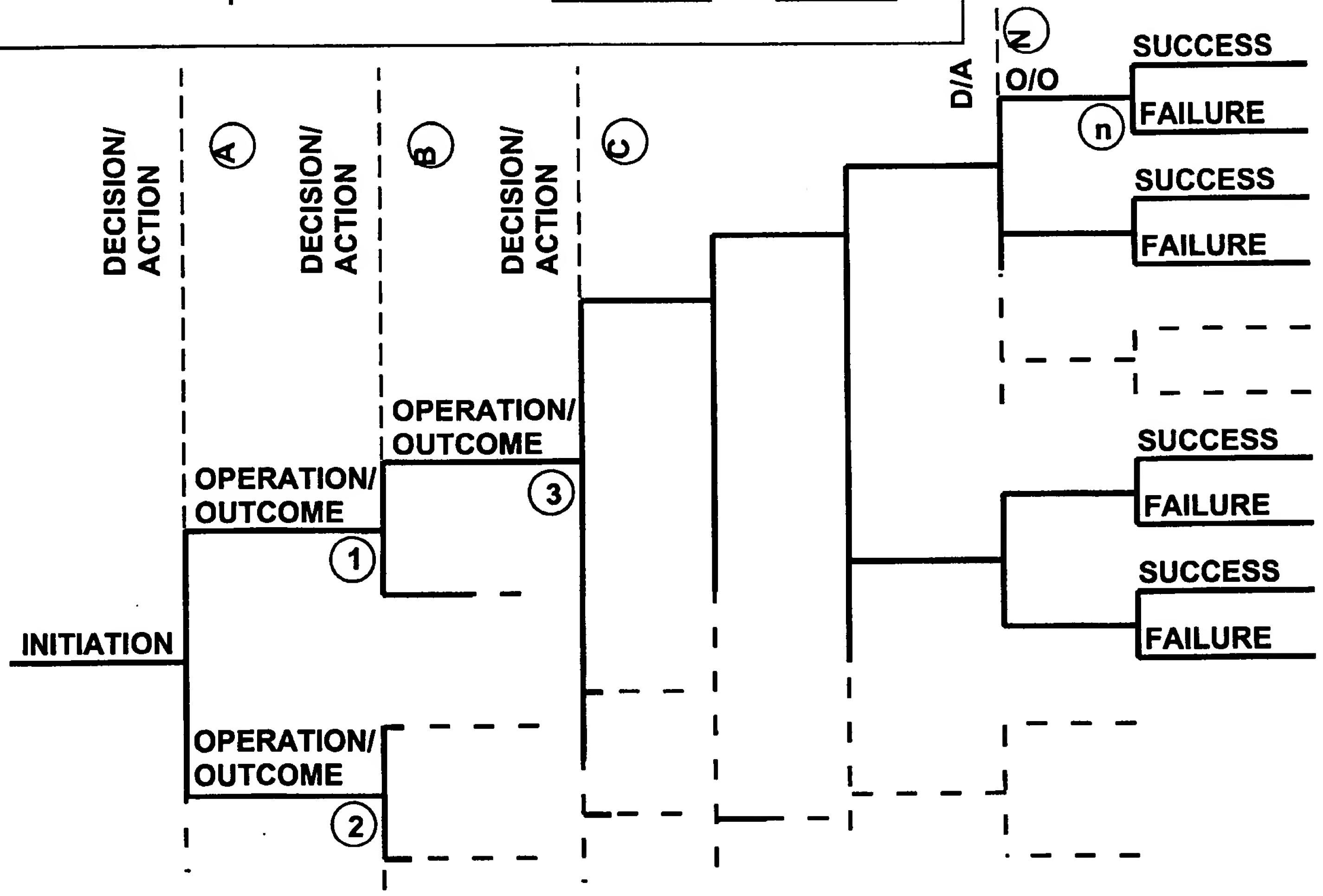
EXAMPLE “CHALLENGES”...

- Pipe or Vessel Burst
- Ignition of Stored Combustibles
- Technology Need
- Utility System Failure
- Outbreak of Epidemic
- Heightened Business Competition
- Normal System Operating Command

provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

EVENT TREE ANALYSIS (General Case)...

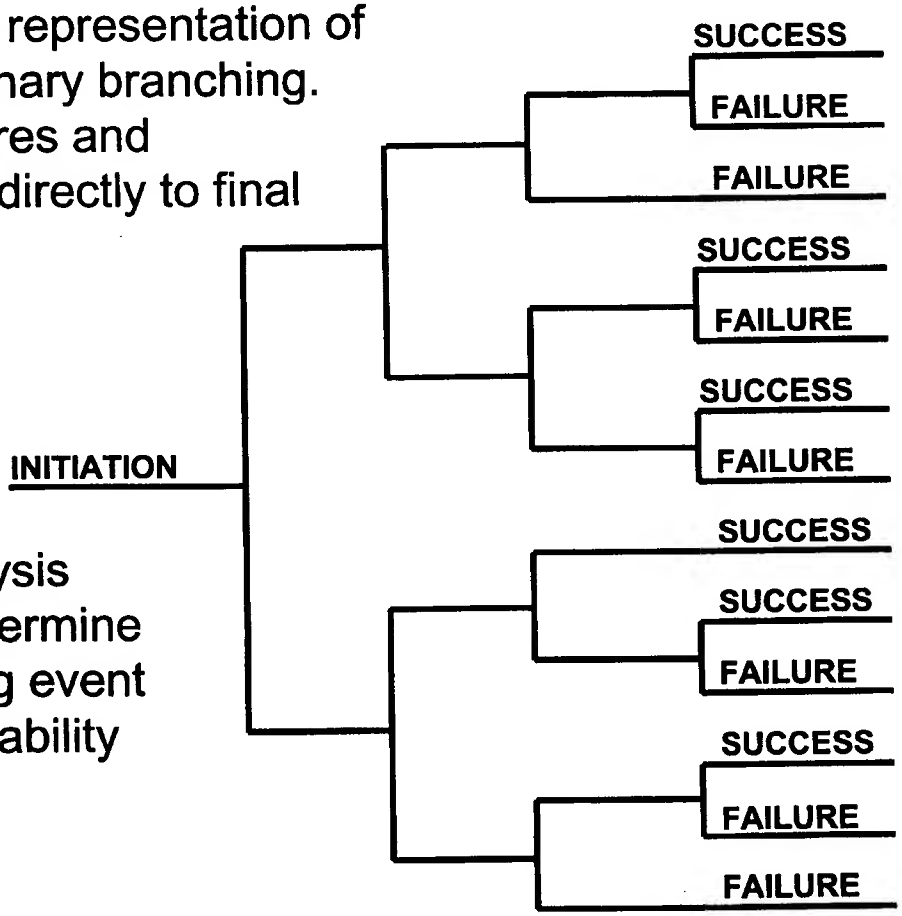
Portray all credible system operating permutations.
Trace each path to eventual success or failure.



provided as a free service by www.fault-tree.net – Pat L. Clemens and Jacobs Sverdrup

EVENT TREE ANALYSIS (Bernoulli Model)...

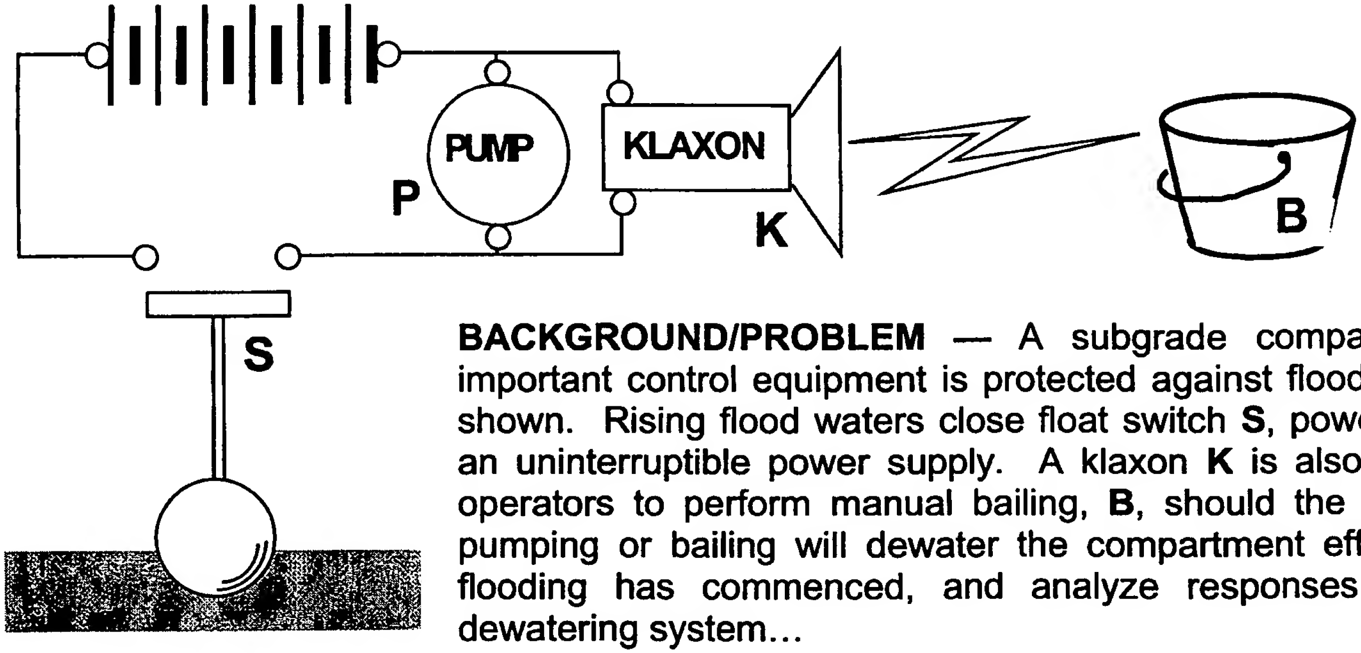
Reduce tree to simplified representation of system behavior. Use binary branching. Lead unrecoverable failures and undefeatable successes directly to final outcomes.



A fault tree or other analysis may be necessary to determine probability of the initiating event or condition. (Unity probability may be assumed.)

provided as a free service by www fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

AN EXAMPLE PROBLEM...



BACKGROUND/PROBLEM — A subgrade compartment containing important control equipment is protected against flooding by the system shown. Rising flood waters close float switch **S**, powering pump **P** from an uninterruptible power supply. A klaxon **K** is also sounded, alerting operators to perform manual bailing, **B**, should the pump fail. Either pumping or bailing will dewater the compartment effectively. Assume flooding has commenced, and analyze responses available to the dewatering system...

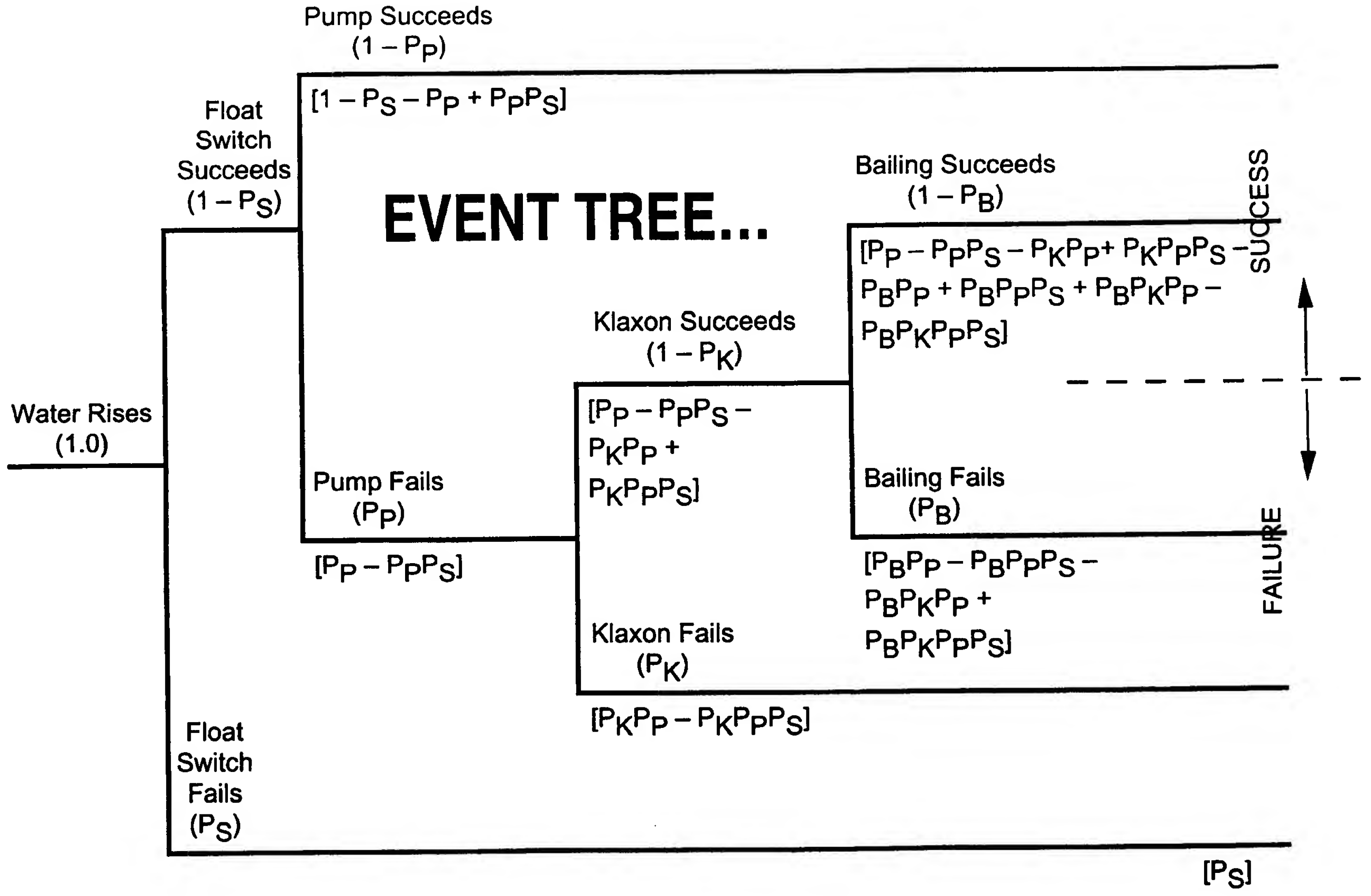
- Develop an event tree representing system responses.
- Develop a reliability block diagram for the system.
- Develop a fault tree for the TOP event *Failure to Dewater*.

SIMPLIFYING ASSUMPTIONS:

- Power is available full time.
- Treat only the 4 system components **S**, **P**, **K**, and **B**.
- Consider operator error as included within the bailing function, **B**.

provided as a free service by www.fault-tree.net – Pat L. Clemens and Jacobs Sverdrup

EVENT TREE...



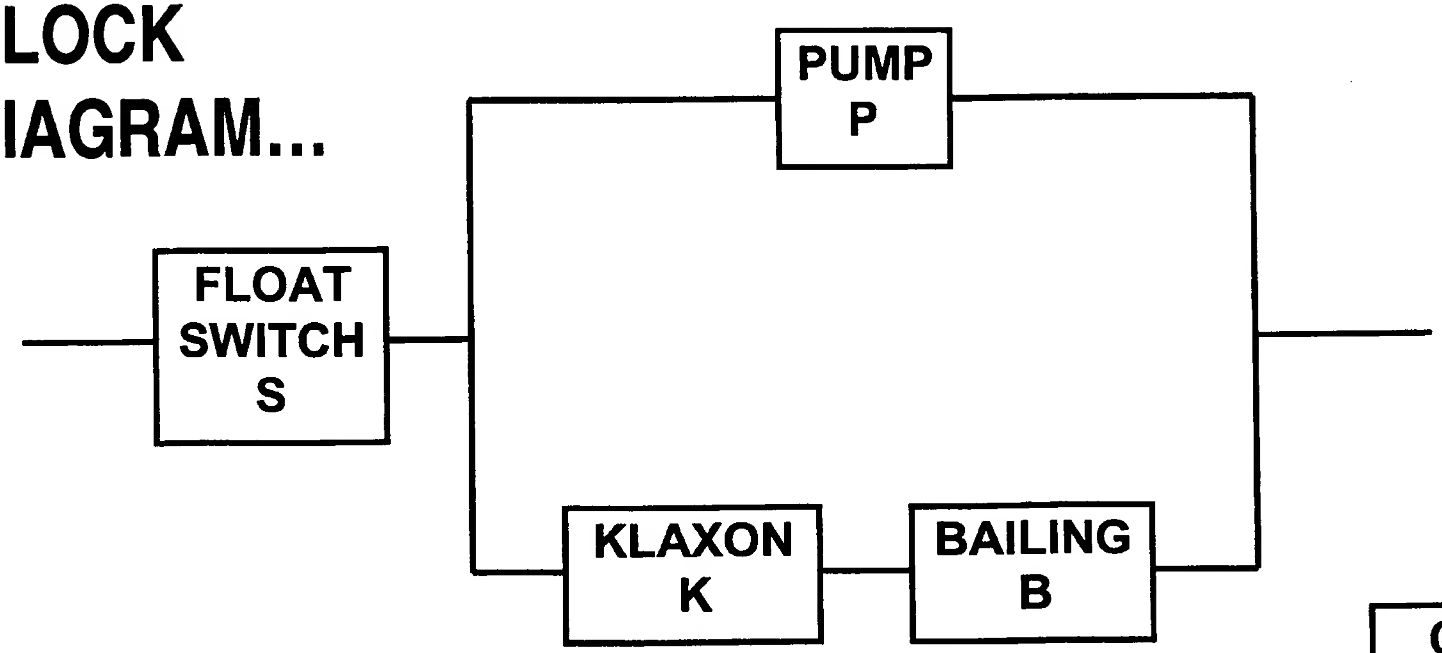
$$P_{SUCCESS} = 1 - P_S - P_K P_P + P_K P_P P_S - P_B P_P + P_B P_P P_S + P_B P_K P_P - P_B P_K P_P P_S$$

$$P_{FAILURE} = P_S + P_K P_P - P_K P_P P_S + P_B P_P - P_B P_P P_S - P_B P_K P_P + P_B P_K P_P P_S$$

$$P_{SUCCESS} + P_{FAILURE} = 1$$

provided as a free service by www.fault-tree.net – Pat L. Clemens and Jacobs Sverdrup

RELIABILITY BLOCK DIAGRAM...



PATH SETS
S / P
S / K / B

CUT SETS
S
P / K
P / B

provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

FAULT TREE...

<u>EXACT SOLUTION</u>

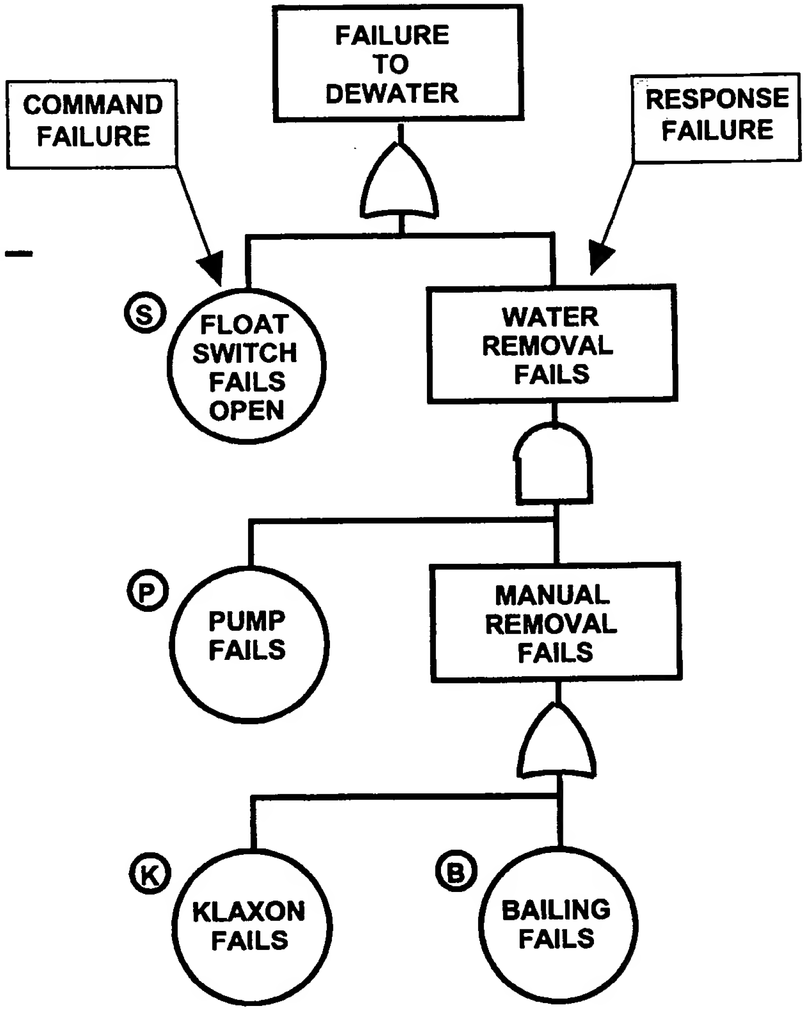
$P_{TOP} = P_S + P_P P_K - P_P P_K P_S + P_B P_P - P_B P_P P_S - P_B P_K P_P + P_B P_K P_P P_S$

<u>RARE EVENT APPROXIMATION</u>

$P_{TOP} = P_S + P_P P_K + P_P P_B$

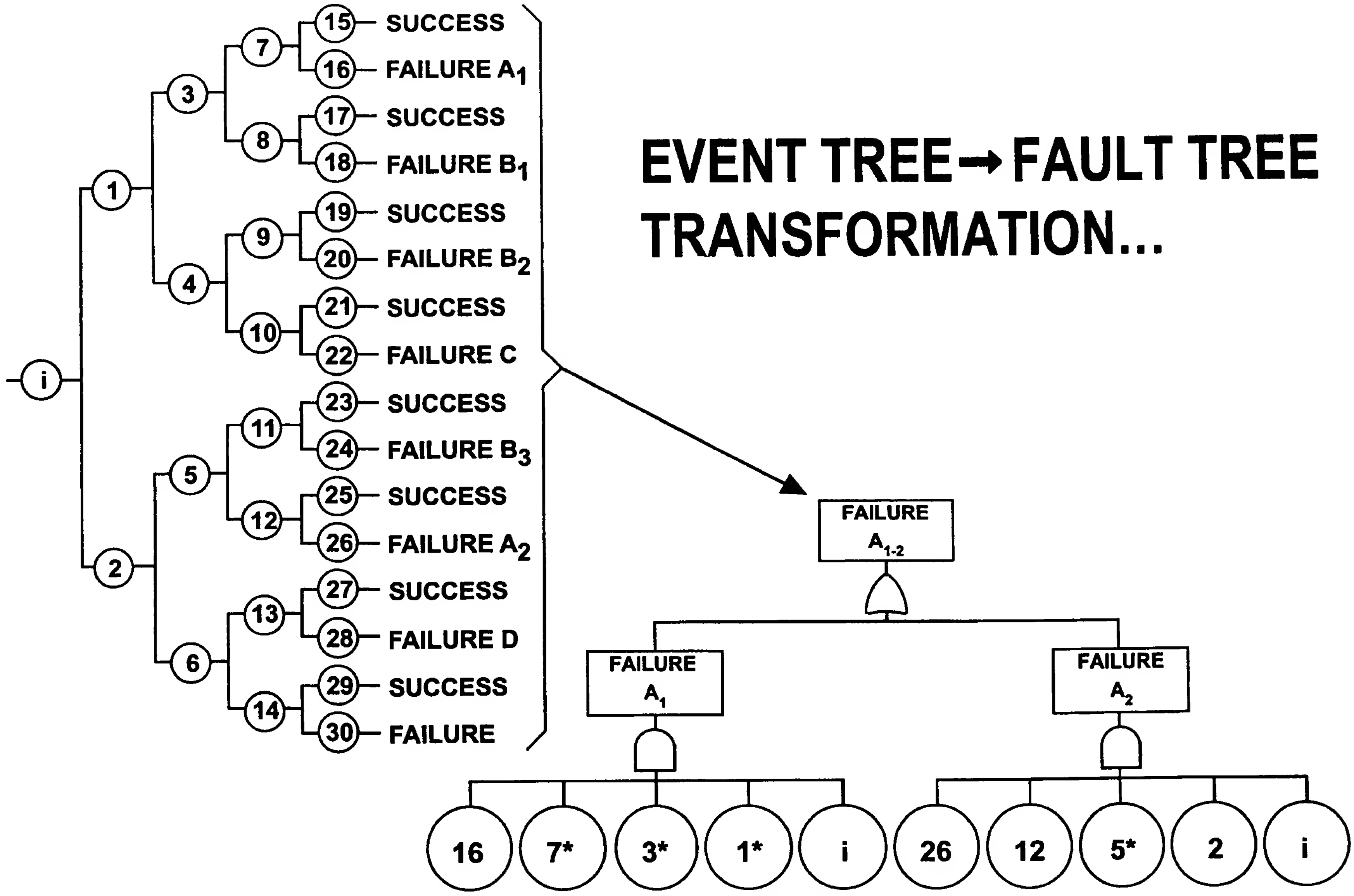
PATH SETS
S / P
S / K / B

CUT SETS
S
P / K
P / B



provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

EVENT TREE→FAULT TREE TRANSFORMATION...



*Note that not all events represented here are failures.

provided as a free service by www.fault-tree.net – Pat L. Clemens and Jacobs Sverdrup

ASSESS RISK AND JUDGE TOLERABILITY...

Failure statements express SEVERITY

Event Tree Analysis explores OUTCOMES / assesses PROBABILITY

PROBABILITY and SEVERITY establish RISK

IS THE RISK ACCEPTABLE?

If not, develop intervenors!

Select intervenor(s) on the basis of:

EFFECTIVENESS

COST

FEASIBILITY (incl. schedule)

provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

EVENT TREE SHORTCOMINGS & ADVANTAGES...

SHORTCOMINGS:

Operating pathways must be anticipated.

Partial successes/failures are not distinguishable.

Initiating events are treated singly. (Multiple trees are required for multiple events; co-existing initiating events are not considered.)

- Sequence-dependent scenarios are not modeled well.

- ADVANTAGES:

End events need not be foreseen.

- Multiple failures can be analyzed.

Potential Single-Point Failures can be identified.

System weaknesses can be identified.

- Zero-payoff system elements/options can be discarded.

BIBLIOGRAPHY — Selected references for further study...

- Center for Process Safety; “Guidelines for Hazard Evaluation Procedures; 2nd Edition with Worked Examples” 1992 (461 pp); American Institute of Chemical Engineers
- Lees, Frank P.; “Loss Prevention in the Process Industries”; 1980 (1316 pp — two volumes)
- Henley, Ernest J. & Hiromitsu Kumamoto; “Reliability Engineering and Risk Assessment”; 1981 (568 pp)

provided as a free service by www.fault-tree.net -- Pat L. Clemens and Jacobs Sverdrup

Appendix B

Fault Tree Analysis – A History

Clifton A. Ericson II
The Boeing Company; Seattle, Washington

ABSTRACT

Fault Tree Analysis (FTA) is a tool for analyzing, visually displaying and evaluating failure paths in a system, thereby providing a mechanism for effective system level risk evaluations. Many people and corporations are already familiar with this tool and use it on a regular basis for safety and reliability evaluations. In some fields it is required for product certification.

FTA is now about 39 years old, and has become a well-recognized tool worldwide. Many improvements have been made since the inception of FTA in 1961 and many people have been involved. This paper provides an overview on the historical aspects of the FTA industry. Topics include important developments through the years, improvements in the process, people involved and contributions made.

FTA has become an important tool in systems design and development, and its history should be recorded and the appropriate people duly recognized. The intent of this paper is to provide a fuller appreciation of the people and events that have contributed to the development of FTA. This is a necessarily incomplete, but hopefully, representative survey of the events, people and literature which have become associated with FTA.

FUNDAMENTAL CONCEPT OF FTA

The fundamental concept of Fault Tree Analysis is the translation of the failure behavior of a physical system into a visual diagram and logic model. The diagram segment provides a visual model that very easily portrays system relationships and root cause fault paths. The logic segment of the model provides a mechanism for qualitative and quantitative evaluation. FTA is based on Reliability theory, Boolean algebra and probability theory. A very simple set of rules and symbols provides the mechanism for analyzing very complex systems, and complex relationships between hardware, software and humans.

EARLY HISTORY

H. A. Watson of Bell Laboratories in connection with an U.S. Air Force contract to study the Minuteman Launch Control System [ref. 1] first conceived fault Tree Analysis. Dave Haasl, then at the Boeing Company, recognized the value of this tool and led a team that applied FTA to the entire Minuteman Missile System. Other divisions within Boeing saw the results from the Minuteman program and began using FTA during the design of commercial aircraft. In 1965 Boeing and the University of Washington sponsored the first System Safety Conference. At this conference, several papers were presented on FTA, marking the beginning of worldwide interest in FTA.

In 1966 Boeing developed a simulation program called BACSIM for the evaluation of multi-phase fault trees. BACSIM could handle up to 12 phases, and included the capability for repair and K-factor adjustment of failure rates. Boeing also developed a program that plotted fault trees on a Calcomp 26-inch wide roll plotter. Both programs ran on an IBM 370 mainframe. These were in-house Boeing programs, developed by Bob Schroeder, that few people were aware of outside Boeing.

Following the lead of the aerospace industry, the nuclear power industry discovered the virtues and benefits of FTA, and began using the tool in the design and development of nuclear power plants. Many key individuals in the nuclear power industry contributed to advancing fault tree theory and fault tree software codes. In fact, the nuclear power industry may have contributed more to the development of FTA than any other single user group. Many new evaluation algorithms were developed, along with software using these algorithms.

FTA has also been adopted by the chemical process industry, the auto industry, rail transportation and is now starting to be utilized by the robotics industry. There are probably many other industries and disciplines using FTA that have not been mentioned here.

UNWANTED RECOGNITION

As is sometimes the case in system safety, a project is not given adequate safety attention until after an accident or incident has occurred. Then, system safety is rigorously applied to solve the accident problem, and any others that might be lurking in the woodwork. The following three accidents were unfortunate and unwanted, but they helped to better establish the FTA process.

After the Apollo 1 launch pad fire on January 27, 1967, NASA hired Boeing to implement an entirely new and comprehensive safety program for the entire Apollo project. As part of this safety effort, fault tree analysis was performed on the entire Apollo system, which helped to bring FTA into national limelight.

Following the Three Mile Island nuclear power plant accident on March 28, 1979, several accident review studies were conducted utilizing FTA. Several years prior to this accident the WASH-1400 study (1976) was conducted to review nuclear power plant design, and to assure the public that the probability of nuclear accidents was very small. This study used fault tree analysis quite extensively, which helped to legitimize the tool and promote its use in the accident investigation..

The Space shuttle Challenger accident occurred on January 28, 1986. Following this accident an independent review team used fault trees to evaluate the main engines to ensure adequate safety in the design. This study showed the applied benefits of FTA.

SUMMARY OF EVENTS

The Beginning Years (1961 – 1970)

1. H. Watson of Bell Labs, along with A. Mearns, developed the technique for the Air Force for evaluation of the Minuteman Launch Control System, circa 1961.
2. Recognized by Dave Haasl of Boeing as a significant system safety analysis tool (1963).
3. First major use when applied by Boeing on the entire Minuteman system for safety evaluation (1964 – 1967, 1968-1999).
4. The first technical papers on FTA were presented at the first System Safety Conference, held in Seattle, June 1965.
5. Boeing began using FTA on the design and evaluation of commercial aircraft, circa 1966.
6. Boeing developed a 12-phase fault tree simulation program, and a fault tree plotting program on a Calcomp roll plotter.

The Early Years (1971 – 1980)

1. Adopted for use by the Nuclear Power industry.
2. Many new evaluation algorithms were developed.
3. Many new fault tree evaluation software codes were developed. Some of the more recognized software includes Prepp/Kitt, SETS, FTAP, Importance and COMCAN.

The Mid Years (1981 – 1990)

1. Usage started becoming international, primarily via the Nuclear Power industry.
2. More evaluation algorithms and codes were developed.
3. A large number of technical papers were written on the subject.
4. Usage of FTA in the software (safety) community.

The Present (1991 – 1999)

1. Continued use on many systems in many countries.
2. High quality fault tree construction and evaluation software developed that operates on PC's.
3. Usage of FTA adopted by the Robotics industry.

HIGHLIGHTS

The following provides some of the highlights of individuals and their contributions over the years. Dave Haasl devised a construction methodology and construction rules that have been followed almost implicitly by everyone in the industry [training classes and Table 3-1]. Jerry Fussell initiated automatic FT construction with his Synthetic Tree Model (STM) [Table 5-3]. Powers and Tompkins developed an automated fault tree construction method for chemical systems [Table 2-9]. Lapp and Powers developed the Fault Tree Synthesis (FTS) program, which utilizes a di-graph model [Table 2-12]. In 1970 W. Vesely developed the Kinetic Tree Theory (KITT) and the PREPP/KITT computer program [Table 4-1 and 4-3]. Fussell and Vesely developed the top down cut set generation algorithm called MOCUS (Method of obtaining Cut Sets) [Table 5-1 and 5-4]. A bottom up cut set algorithm call MICSUP (Minimal Cut Sets Upward) was developed by P. Pande, M. Spector and P. Chatterjee [Table 2-11]. The FATRAM algorithm was developed by D. Rasmuson and N. Marshall to improve upon the MOCUS algorithm [Table 2-14]. S. Semanderes developed a cut set algorithm using prime numbers in his ELRAFT program, which efficiently stored cut sets and eliminated super sets. [Table 2-8]. Randall Willie developed the ever-popular computer program call FTAP [Table 2-13]. Dick Worrell developed the SETS computer program, which is still in usage in various versions [Table 2-10]. Howard Lambert extensively developed importance measures, and developed the program called IMPORTANCE [Table 6-4, 6-7].

IMPORTANT CONTRIBUTORS

Through out the years many individuals have contributed to the development of FTA, some more than others. The following is a short list of some of the individuals who have made significant contributions to the field. This list is derived from both the literature available and personal knowledge. Although this list is not complete, it is an attempt to recognize th se who made valuable contributions. Each of these individuals has carried the banner and been a major spokesperson for FTA.

1. H. Watson and A. Mearns
   Developed FTA methodology [Table 2-1, 2-2]
2. Dave Haasl
   Developed FT construction techniques, training [Table 2-3, 3-1]
3. Robert Schroeder
   BACSIM simulation program, AFTD program (Boeing)
4. William Vesely
   Kinetic tree theory, Kitt/Prepp, technical papers from 1969-1994 [Table 4]
5. Jerry Fussel
   Synthetic FTA, MOCUS, technical papers from 1972-1994 [Table 5]
6. Howard Lambert
   Developed Importance program, technical papers from 1973-1994 [Table 6]
7. Dick Worrel
   SETS program [Table 1-10]
8. Randall Willie
   FTAP program [Table 1-13]
9. Ernst Henely
   Technical papers and books from 1973-1996 [Table 7]
10. John Andrews
    Research, technical papers and books from 1986-1999 [Table 8]

PRODUCT APPLICATIONS

Since its inception, FTA has been applied to many different types of systems and hardware, on many different projects. The following list shows many areas that have used FTA. This list may not be complete, but it is fairly representative of systems that have received FTA.

Major industries and technologies utilizing FTA include:
1. Aircraft – commercial, fighters, bombers, tankers, UAV's, AWACS, helicopters
2. Power Systems – nuclear, solar, electric
3. Transit Systems – trains, MPRT (Morgantown Personal Rapid Transit), BART
4. Space –Apollo, Space Shuttle, satellites, launch vehicles, Space Station
5. Robotic Systems
6. Auto Systems
7. Missile Systems – Minuteman, SRAM, ALCM, Tomahawk
8. Oil Platforms
9. Torpedoes
10. Hydrofoil

APPLICATION PURPOSES

FTA tends to be used in high-risk applications where a probability of occurrence assessment is needed. However, FTA has proven its worth for both quantitative and qualitative applications. Some of the most typical reasons why FTA has been used include the following.

Major applications of FTA include:
1. Numerical requirement verification
2. Identification of safety critical components
3. Product certification
4. Product risk assessment
5. Accident/incident analysis
6. Design change evaluation
7. Visual diagrams of cause-consequence events
8. Common cause analysis

MAJOR IMPROVEMENTS

FTA construction and evaluation is a relatively simple and straightforward process. However, when trees become large and complex, they become much more difficult to solve. The ability to evaluate fault trees is directly correlated to size, complexity and computer capability. In the early day's, computer power and capacity was much more limited than it is today, which meant that much research went into developing tractable FTA algorithms. Many algorithms and computer software codes were developed to efficiently take advantage of various FT parameters.

Early FT computer software was limited by computer power – memory and speed. Current codes have the advantage of a) early algorithms and b) improvements in computer power. FTA evaluation programs have become decentralized and user friendly. They now operate on PC's sitting on an analyst's desk, rather than a mainframe computer.

Probably the two most major advances in FTA technology are 1) improved user interface, and 2) improved user computational power. And, both of these advances are the direct result of improvements in computer technology. If the computer industry had not improved, these gains would not likely have been achieved for FTA.

Major improvements in the FTA process include:
1. Progressed from drawing trees manually using templates to using computers
2. Transition from mainframe computers to desktop PC's

3. Transition from large mainframe plotters to desktop printers
4. Decentralized – moved the computer and FTA tools to the user's desk
5. Software has become user friendly
6. Graphical user interface rather than ASCII text files with no visuals
7. Software packages have become relatively inexpensive
8. Improved computation – algorithms, size, speed

PHILOSOPHIES AND TRENDS

Over the years there have been various trends and philosophies in FTA, some have come and gone, others are still with us. Some of these trends have even competed with each other.

1. Analytical solution vs. Simulation
2. Top down vs. Bottom up algorithms
3. Dynamic FTA vs. Static FTA
4. SFTA (Software FTA)
5. Fuzzy FTA
6. FT Synthesis (automated FT construction)
7. Single phase vs. multi phase fault trees
8. Various evaluation methods
   - Boolean reduction
   - BDD (Binary Decision Diagram)
   - Min terms
   - Genetic algorithms
   - Approximations

TECHNICAL ARTICLES

FTA is a topic that almost everyone wants to write about, and almost everyone feels qualified to do so. To date I have cataloged 775 technical articles and books on the subject of FTA. The list of authors ranges from experienced practitioners, to dedicated researchers, to neophytes that just learned how to perform FTA the week before. Table 1 shows a graph of the number of items on FTA per year since its inception. The quantity of items indicates that FTA is an important and valuable subject, with widespread interest. This chart shows the continued, and somewhat constant, interest in FTA. Table 2 lists some of the early articles on FTA. Table 3 lists the most used books on FTA. Tables 4 through 8 contain technical articles written by five different authors. Each of these authors have individually contributed the most research and technical articles on FTA, and have proudly carried the FTA banner for many years. Although other individuals have made important contributions, these five have been the most consistent.

SUMMARY

FTA was conceived circa 1961, and as such is a relatively new tool compared to many other technical tools and disciplines. Special recognition should go to H. A. Watson as the Father of Fault Tree Analysis and Dave Haasl as the God Father of Fault Tree Analysis. Watson of Bell Labs invented fault tree analysis (along with assistance from M. A. Mearns). Haasl, while at Boeing saw the benefits of FTA and spearheaded the first major application on the Minuteman program. As a private consultant he has helped train most of the industry in FTA, and consulted on many projects utilizing FTA.

Algorithms and software codes received the most research attention in the early years (and still does). Less research has been spent on improving construction methods and training methods. Advancements and improvements in computer technology have provided concomitant advances in FTA. FTA analysts can today visually construct fault trees on desktop computers with relatively inexpensive software

Synthetic FTA is the process of using computers to automatically construct fault trees from electrical schematics and drawings. Synthetic FTA has been an elusive goal. Researchers have been trying to achieve this objective since 1970, yet there are still no commercial products available.

FTA has earned its place as a valuable tool for safety, risk assessment, accident investigation, reliability, etc. The number of papers graphed over time show that the interest in FTA has not declined, but has actually remained constant over the years. There have been criticisms of FTA over the years, but the benefits and strengths of FTA have proven to out weight the detractors arguments, and FTA has become an internationally recognized and used tool.

BIOGRAPHY

Clifton A. Ericson II
The Boeing Company
18247 150th Ave SE
Renton, WA 98058 USA
phone 253-657-5245
fax 253-657-2585
email clifton.a.ericson@boeing.com

Mr. Ericson works in system safety on the Boeing 767 AWACS program. He has 34 years experience in system safety and software design with the Boeing Company. He has been involved in all aspects of fault tree development since 1965, including analysis, computation, multi-phase simulation, plotting, documentation, training and programming. He has performed Fault Tree Analysis on Minuteman, SRAM, ALCM, Apollo, Morgantown Personal Rapid Transit, B-1, AWACS and 737/757/767 systems. He is the developer of the MPTREE, SAF and FTAB fault tree computer programs. In 1975 he helped start the software safety discipline, and has written papers on software safety and taught software safety at the University of Washington. Mr. Ericson holds a BSEE from the University of Washington and an MBA from Seattle University. He is currently Executive Vice President of the System Safety Society, and is on the technical review committee for the Hazard Prevention journal.

Table 1—Fault Tree Articles Per Year

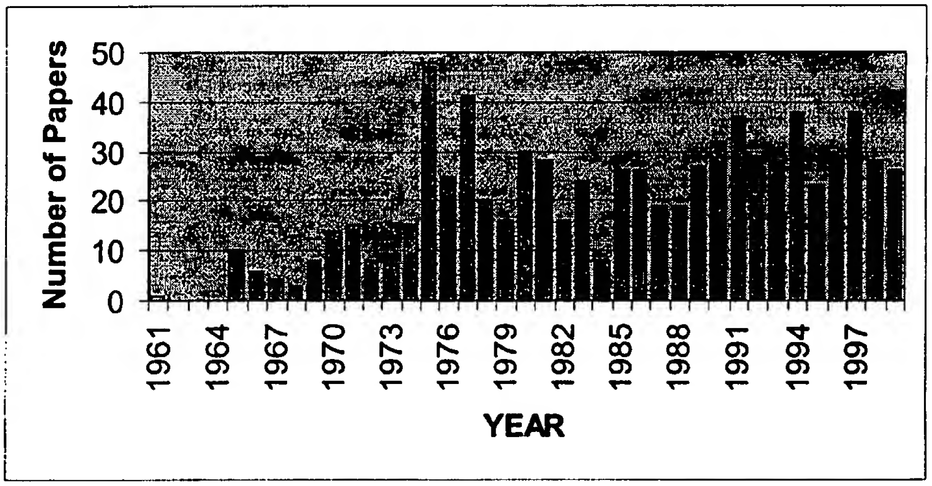


Table 2 – Early Works

	Year	Title
1.	1961	*Launch Control Safety Study*, Section VII Vol 1; Bell Labs; Murray Hill, NJ, 1961, H. A. Watson
2.	1965	*Fault Tree Analysis : The Study Of Unlikely Events In Complex Systems*, Boeing/UW System Safety Symposium, 1965, A. B. Mearns
3.	1965	*Advanced Concepts In Fault Tree Analysis*, Boeing/UW System Safety Symposium, 1965, D. F. Haasl
4.	1965	*Computer Evaluation Of The Safety Fault Tree Model*, System Safety Symposium (Boeing/UW), 1965, J. M. Michels
5.	1965	*A Monte Carlo Method To Compute Fault Tree Probabilities*, System Safety Symposium (Boeing/UW), 1965, P. M. Nagel
6.	1965	*The Application of Fault Tree Analysis to Dynamic Systems,* System Safety Symposium (Boeing/UW), 1965, R. J. Feutz & T. A. Waldeck
7.	1965	*Concept of System Safety Mathematics*, System Safety Symposium (Boeing/UW), 1965, K. Kanda
8.	1971	*ELRAFT: A Computer Program For The Efficient Logic Reduction Analysis Of Fault Trees*, 1971, IEEE Transactions On Nuclear Science (NS-18 No 1), p481-487, S. N. Semanderes
9.	1974	*Fault Tree Synthesis for Chemical Processes*, G. J. Powers and F. C. Tompkins, AICHE Journal, Vol 20, 1974, p376-387
10.	1974	*Set Equation Transformation System (SETS)*, 1974, SLA-73-0028A Sandia National Laboratories, R. B. Worrell
11.	1975	*Computerized Fault Tree Analysis: TREEL and MICSUP*, 1975, Univ. of California ORC-75-3, P. K. Pande & M. E. Spector & P. Chatterjee
12.	1977	*The Synthesis of Fault Trees*, S. A. Lapp and G. J. Powers, in Nuclear Systems Reliability and Risk Assessment , 1977, p778-799
13.	1978	*Computer Aided Fault Tree Analysis: FTAP*, 1978, Univ. of California Operations Research Center; OC 78-14, R. R. Willie

14.	1978	*FATRAM – A Core Efficient Cut Set Algorithm*, 1978, IEEE Transactions On Reliability (R-27 No 4), p250-253, D. M. Rasmuson & N. H. Marshall

Table 3 – Most Significant Books

	Year	Title
1.	1981	*Fault Tree Handbook*, NUREG-0492, 1981, N. H. Roberts, W. E. Vesely, D. F. Haasl & F. F. Goldberg
2.	1993	*Reliability and Risk Assessment*, Longman Scientific & Technical, 1993, J. D. Andrews & T. R. Moss
3.	1996	*Probabilistic Risk Assessment And Management For Engineers And Scientists*, IEEE Press (2nd edition), 1996, E. J. Henley & H. Kumamoto

Table 4 – Articles by W. E. Vesely

	Year	Title
1.	1969	*Analysis Of Fault Trees By Kinetic Tree Theory*, Idaho Nuclear Corp IN-1330, 1969, W. E. Vesely
2.	1970	*A Time- Dependent Methodology For Fault Evaluation*, Nuclear Engineering and Design (Vol 13 No 2), 1970, p337-360, W. E. Vesely
3.	1970	*PREPP & KITT: Computer Codes For The Automatic Evaluation Of A Fault Tree*, IN-1349; Idaho Nuclear Corp, 1970, W. E. Vesely & R. E. Narum
4.	1971	*Reliability And Fault Tree Applications At The NRTS*, IEEE Transactions On Nuclear Science (NS-18 No 1), 1971, p472-480, W. E. Vesely
5.	1972	*A New Methodology For Obtaining Cut Sets For Fault Trees*, Transactions American Nuclear Society (Vol 15 No 1), 1972, p262-263, J. B. Fussell & W. E. Vesely
6.	1975	*Reliability Quantification Techniques Used In The Rasmussen Study*, Reliability And Fault Tree Analysis: SIAM, 1975, p775-804, W. E. Vesely
7.	1976	*Important Event Tree And Fault Tree Considerations In The Reactor Safety Study*, IEEE Transactions On Reliability (R-25 No 3, 1976, p132-139, S. Levine & W. E. Vesely
8.	1977	*FRANTIC – A Computer Code For Time dependent Unavailability Analysis*, NUREG 0193, 1977, W. E. Vesely & F. F. Goldberg
9.	1983	*The Façade Of Probabilistic Risk Analysis: Sophisticated Computation Does Not Necessarily Imply Credibility*, Proceedings Annual R & M Symposium, 1983, p49-51, W. E. Vesely
10.	1985	*Two Measures of Risk Importance and their Application*, Nuclear Technology (Vol 68 No 2), 1985, p226-234, W. E. Vesely
11.	1988	*Utilizing Probabilistic Risk Analyses (PRA) in Decision Support Systems*, Engineering Risk and Hazard Assessment; Volume II; editors A. Kandel & E. Avni; CRC Press, 1988, p101-116, W. E. Vesely
12.	1994	*PRA Importance Measures For Maintenance Prioritization Applications*, Reliability Engineering And System Safety 43, 1994, p307-318, W. E. Vesely, M. Belhadj & J. T. Rezos

Table 5 – Articles by J. B. Fussell

	Year	Title
1.	1972	*A New Methodology For Obtaining Cut Sets For Fault Trees*, Transactions American Nuclear Society (Vol 15 No 1), 1972, p262-263, J. B. Fussell & W. E. Vesely
2.	1973	*A Formal Methodology For Fault Tree Construction*, Nuclear Science and Engineering (Vol 52), 1973, p421-432, J. B. Fussell
3.	1973	*Synthetic Tree Model – A Formal Methodology For Fault Tree Construction*, ANCR-1098, 1973, J. B. Fussell

4.	1974	*MOCUS – A Computer To Obtain Minimal Sets From Fault Trees*, Aerojet Nuclear Corp; ANCR-1156, 1974, J. B. Fussell & E. B. Henry & N. H. Marshall
5.	1974	*Fault Trees – A State Of The Art Discussion*, IEEE Transactions On Reliability (Vol R-23 No 1), 1974, p51-55, J. B. Fussell & G. J. Powers & R. G. Bennetts
6.	1975	*Fault Tree Analysis – Concepts And Techniques*, NATO Advanced Study Institute On Generic Techniques Of System Reliability Assessment; Nordhoff Netherlands, 1975, J. B. Fussell
7.	1975	*How to Hand Calculate System Reliability Characteristics*, IEEE Transactions On Reliability (R-24 No3), 1975, p169-174, J. B. Fussell
8.	1975	*Reliability And Fault Tree Analysis*, Conference On Reliability And Fault Tree Analysis; UC Berkeley; SIAM Pub, 1975, R. E. Barlow & J. B. Fussell & N. D. Singpurwalla
9.	1975	*Computer Aided Fault Tree Construction For Electrical Systems*, Reliability And Fault Tree Analysis ; SIAM, 1975, p37-56, J. B. Fussell
10.	1975	*Fault Tree Analysis – The Secondary Failure Anomaly*, Operations Research Society Of America, 1975, J. B. Fussell
11.	1976	*Fault Tree Analysis : Concepts And Techniques*, Generic Techniques In Systems Reliability Assessment; E.J.Henley & J.W.Lynn editors; Noordhoff Pub, 1976, p133-162, J. B. Fussell
12.	1976	*A Collection Of Methods For Reliability And Safety Engineering*, ANCR-1273; Idaho National Engineering Lab, 1976, J. B. Fussell & G. R. Burdick & D. M. Rasmuson & J. C. Wilson
13.	1976	*On The Quantitative Analysis Of Priority AND Failure Logic*, IEEE Transactions On Reliability (R-25 No 5), 1976, p324-326, J. B. Fussell & E. F. Aber & R. G. Rahl
14.	1976	*Quantitative Evaluation Of Nuclear System Reliability And Safety Characteristics*, IEEE Transactions On Reliability (R-25 No3), 1976, p178-183, J. B. Fussell & H. E. Lambert
15.	1977	*Nuclear Systems Reliability and Risk Assessment*, SIAM Pub; International Conference on Nuclear Systems Reliability Engineering and Risk Assessment, 1977, J. B. Fussell & G. R. Burdick
16.	1977	*Common Cause Failure Analysis Methodology For Complex Systems*, Nuclear Systems Reliability and Risk Assessment; edited by J. B. Fussell & G. R. Burdick; SIAM, 1977, p289-313, D. P. Wagner & C. L. Cate & J. B. Fussell
17.	1977	*BACFIRE – A Computer Program For Common Cause Failure Analysis*, Univ. Tennessee NERS-77-02, 1977, C. L. Cate & J. B. Fussell
18.	1977	*Phased Mission Analysis: A Review Of New Developments And An Application*, IEEE Transactions On Reliability (Vol R-26 No 1), 1977, p43-49, G. R. Burdick & J. B. Fussell & D. M. Rasmuson & J. R. Wilson
19.	1977	*Fault Tree Analysis As A Part Of Mechanical System Design*, National Bureau Of Standards NBS-SP-487 NTIS, 1977, J. B. Fussell & D. P. Wagner
20.	1980	*A Methodology for Calculating the Expected Number of Failures of a System Undergoing a Phased Mission*, Nuclear Science Engineering (Vol 74), 1980, D. F. Montague & J. B. Fussel
21.	1981	*System Reliability Engineering Methodology For Industrial Application*, Loss Prevention Vol 14; AICE, 1981, p18-28, J. S. Arendt & J. B. Fussell
22.	1994	*Probabilistic Safety Analysis For Systems With Standby Subsystems With Sequentially Used Standbys*, Reliability Engineering And System Safety 44, 1994, p67-76, Q. Zhang & H. M. Paula & J. B. Fussell

Table 6 – Articles by Howard Lambert

	Year	Title
1.	1973	*System Safety Analysis And Fault Tree Analysis*, UCID-16238; Lawrence Livermore Labs, 1973, H. E. Lambert
2.	1975	*Fault Trees For Decision Making In Systems Analysis*, Lawrence Livermore Labs UCRL-51829; PhD Thesis; Univ. California, 1975, H. E. Lambert
3.	1975	*Introduction To Fault Tree Analysis*, Reliability And Fault Tree Analysis ; SIAM, 1975, p7-36, H. E. Lambert
4.	1975	*Measures Of Importance Of Events And Cut Sets In Fault Trees*, Reliability And Fault Tree Analysis ; SIAM, 1975, p77-100, H. E. Lambert
5.	1976	*Quantitative Evaluation Of Nuclear System Reliability And Safety Characteristics*, IEEE

		Transactions On Reliability (R-25 No3), 1976, p178-183, J. B. Fussell & H. E. Lambert
6.	1977	*Fault Trees For Diagnosis Of System Fault Conditions,* Nuclear Science And Engineering (Vol 62), 1977, p20-34, H. E. Lambert & G. Yadigaroglu
7.	1977	*The IMPORTANCE Computer Code,* UCRL-79269; Lawrence Livermore Lab, 1977, H. E. Lambert & F. M. Gilman
8.	1978	*The Results Of A Directed Graph Fault Tree Assessment Of A MCA System,* UCRL-80802 Lawrence Livermore Labs, 1978, F. M. Gilman & H. E. Lambert & J. J. Lim
9.	1979	*Comments On The Lapp-Powers Computer Aided Synthesis Of Fault Trees,* IEEE Transactions On Reliability (R-28 No 1), 1979, p6-9, H. E. Lambert
10.	1981	*The Use Of The Computer Code IMPORTANCE With SETS Input,* Sandia SAN81-7068; USNRC Report NUREG/CR-1965, 1981, H. E. Lambert & B. J. Davis
11.	1983	*Interval Reliability For Initiating And Enabling Events,* IEEE Transactions On Reliability (R-32 No 2), 1983, p150-163, C. Dunglinson & H. Lambert
12.	1996	*The Impact Of Improved Vehicle Design On Highway Safety,* Reliability Engineering And System Safety 54, 1996, p65-76, J. S. Eisele & Y. Y. Haimes & N. J. Garber & D. Li & J. H. Lambert & P. Kuzminski & M. Chowdhury

Table 7 – Articles by Ernst Henley

	Year	Title
1.	1973	*Generic Techniques In Systems Reliability Assessment,* Proceedings Of The NATO Advanced Study Institute On Generic Techniques In Systems reliability July 1973; Noordhoff Pub, 1976, E. J. Henley & J. W. Lynn
2.	1976	*Systems Analysis By Sequential Fault Trees,* Microelectronics And Reliability 15, 1976, p247-248, E. J. Henley
3.	1976	*Process Failure Analysis By Block Diagrams And Fault Trees,* Industrial & Engineering Chemistry Fundamentals 15, 1976, p128-134, S. Caceres & E. J. Henley
4.	1977	*Comments On: Computer Aided Synthesis Of Fault Trees,* IEEE Transactions On Reliability (R-26 No 5), 1977, 316-318, E. J. Henley & H. Kumamoto
5.	1978	*Top Down Algorithm For Obtaining Prime Implicant Sets Of Non-Coherent Fault Trees,* IEEE Transactions On Reliability (R-27 No 4), 1978, p242-249, H. Kumamoto & E. J. Henley
6.	1980	*Author Reply #2,* IEEE Transactions On Reliability (R-29), 1980, p133-134, H. Kumamoto & E. J. Henley
7.	1980	*Dagger Sampling Monte Carlo For System Unavailability Evaluation,* IEEE Transactions On Reliability (R-29 No 2), 1980, p122-125, H. Kumamoto & K. Tanaka & K. Inoue & E. J. Henley,
8.	1980	*State Transition Monte Carlo For Evaluating Large Repairable Systems,* IEEE Transactions On Reliability (Vol R-29 No 5), 1980, p376-380, H. Kumamoto & K. Tanaka & K. Inoue & E. J. Henley
9.	1980	*Probabilistic Evaluation Of Prime Implicants And Top Events For Non-Coherent Systems,* IEEE Transactions On Reliability (Vol R-29 No 5), 1980, p361-367, T. Inagaki & E. J. Henley
10.	1981	*Reliability Engineering and Risk Assessment,* Prentice Hall Pub, 1981, E. J. Henley & H. Kumamoto
11.	1981	*Signal Flow Based Graphs For Failure Mode Analysis Of Systems With Control Loops,* IEEE Transactions On Reliability (Vol R-30 No 2), 1981, p110-116, H. Kumamoto & E. J. Henley & K. Inoue,
12.	1986	*Automated Fault Tree Synthesis By Disturbance Analysis,* Industrial & Engineering Chemistry Fundamentals 25, 1986, p233-239, H. Kumamoto & E. J. Henley
13.	1988	*On Digraphs – Fault Trees And Cut Sets,* Reliability Engineering And System Safety (Vol 20 No 2), 1988, p35-61, T. Kohda & E. J. Henley
14.	1989	*Finding Modules in Fault Trees,* IEEE Transactions on Reliability (R-38 No 2), 1989, p165-176, T. Kohda & E. J. Henley & K. Inoue
15.	1990	*An Action-Chain Model For The Design Of Hazard-Control Systems For Robots,* IEEE Transactions on Reliability (R-39 No 2), 1990, p151-157, Y. Sato & E. J. Henley & K. Inoue
16.	1995	*Automated Fault Tree Synthesis By Semantic Network Modeling Rule Based Development And*

		Recursive 3-Value Procedure, Reliability Engineering And System Safety (Vol 49 No 2), 1995, p171-188, H. Kumamoto & E. J. Henley
17.	1996	*Probabilistic Risk Assessment And Management For Engineers And Scientists,* IEEE Press (2nd edition), 1996, E. J. Henley & H. Kumamoto

Table 8 – Articles John Andrews

	Year	Title
1.	1986	*Application Of The Digraph Method Of Fault Tree Construction To Process Plant,* Reliability Engineering And System Safety (Vol 14 No 2), 1986, p85-106, J. D. Andrews & J. M. Morgan,
2.	1988	*The Propagation Of Faults In Process Plants: Fault Tree Synthesis For A Butane Vaporiser System (Part 5),* Reliability Engineering And System Safety 23, 1988, p31-49, J. S. Mullhi & M. L. Ang & B. E. Kelly & F. P. Lees & J. D. Andrews
3.	1990	*Application Of The Digraph Method Of Fault Tree Construction To A Complex Control Configuration,* Reliability Engineering And System Safety (Vol 28), 1990, p357-384, J. Andrews & G. Brennan
4.	1991	*Quantitative Safety Assessment of the Ventilation Recirculation System in an Undersea Mine,* Quality And Reliability Engineering International (Vol), 1991, p497-510, J. D. Andrews
5.	1993	*Fault Tree Analysis,* Reliability And Risk Assessment; Longman Scientific & Technical Publishers, 1993, 144-200, J. D. Andrews & T. R. Moss
6.	1994	*Optimal Safety System Design Using Fault Tree Analysis,* Proceedings ImechE (Vol 208), 1994, p123-132, J. D. Andrews
7.	1995	*New Approaches To Evaluating Fault Trees,* Proceedings Of ESREL 95 Conference, 1995, p241-254, R. M. Sinnamon & J. D. Andrews
8.	1996	*Fault Tree Analysis And Binary Decision Diagrams,* Annual R & M Symposium, 1996, p215-222, R. M. Sinnamon & J. D. Andrews
9.	1996	*Improved Efficiency (Accuracy) In Quantitative Fault Tree Analysis,* Proceedings Of 12th ARTS: Manchester, 1996, R. M. Sinnamon & J. D. Andrews
10.	1997	*Optimal Safety System Performance,* Proceedings Annual R & M Symposium, 1997, p76-83, J. D. Andrews & R. L. Pattison
11.	1997	*Computerized Fault Tree Construction For A Train Braking System,* Quality And Reliability Engineering International (Vol 13), 1997, p299-309, J. J. Henry & J. D. Andrews
12.	1997	*Improved Efficiency In Qualitative Fault Tree Analysis,* Quality And Reliability Engineering International (Vol 13), 1997, p293-298, R. M. Sinnamon & J. D. Andrews
13.	1997	*Improved Accuracy In Quantitative Fault Tree Analysis,* Quality And Reliability Engineering International (Vol 13), 1997, p285-292, R. M. Sinnamon & J. D. Andrews
14.	1997	*New Approaches To Evaluating Fault Trees,* Reliability Engineering And System Safety (Vol 58 No 2), 1997, p89-96, R. M. Sinnamon & J. D. Andrews
15.	1997	*A Computerized Fault Tree Construction Methodology,* Journal of Process Mechanical Engineering (Vol 211 No E3), 1997, p171-183, J. D. Andrew & J. J. Henry
16.	1998	*Efficient Basic Event Orderings for Binary Decision Diagrams,* Proceedings Annual Reliability and Maintainability Symposium, 1998, p61-68, J. D. Andrews & L. M. Bartlett
17.	1998	*Analysis of Systems With Standby Redundancy,* Proceedings of the 16th International System Safety Conference, 1998, p80-89, J. D. Andrews & L. M. Ridley

Appendix C

Fault Tree Analysis

Methods and Applications

Course Project

By:

James M. Kelly

INDE 526

Reliability in Product Design and Testing

University of Washington

August 18, 2003

Introduction

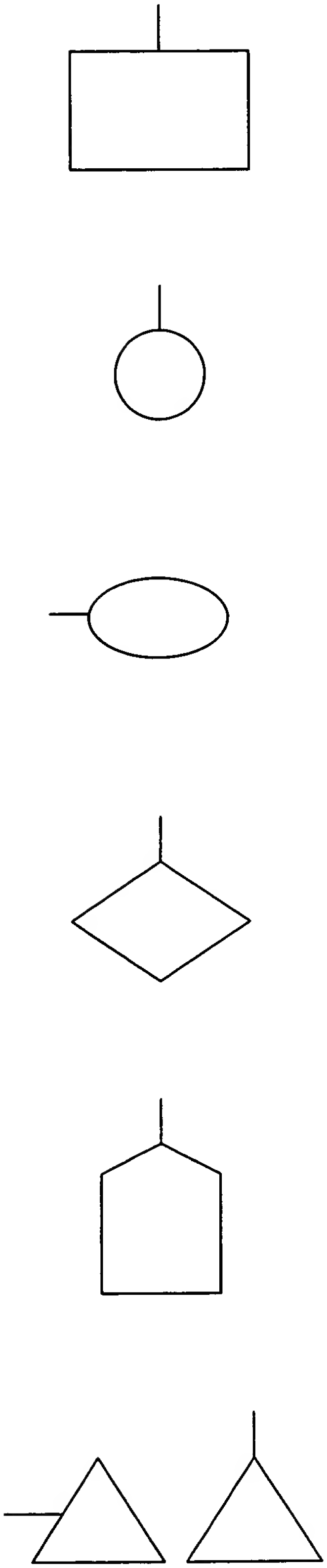
Fault tree analysis was first developed in 1961 by H. A. Watson of Bell Telephone Laboratories to evaluate the safety of the Minuteman Launch Control System [1]. Since that time, it has become a widely used tool for both qualitative and quantitative reliability analysis. In the simplest terms a fault tree is a graphical representation of a system failure mode. This graphical representation illustrates the various parallel and sequential combinations of component failures that combine to result in failure of the system. The system failure mode being considered is usually termed the *"top event"*, and the fault tree is developed in branches below this event. Each branch leads to lower level events that cause the higher level event. Each event in the fault tree is continually redefined in terms of lower level events. The process is terminated when component level failures *or "basic events"* are encountered. These basic events define the *"limit of resolution"* for the analysis. Event levels are separated by logical gates which indicate how the lower level events must combine to initiate the higher level event.

This method of analysis is often referred to as a *"top down approach"* or deductive reasoning. The analysis starts with an assumed failure mode and asks the question *"what can cause this?"* This is in contrast to inductive methods, such as Failure Modes and Effects Analysis (FMEA), which starts with an assumed component failure and determines the effect of the component failure on the system. FMEA asks the question *"what happens if"* this component fails and is often referred to as a *"bottom up approach."*

Basic Building Blocks of Fault Tree Analysis

A fault tree diagram consists of two basic elements, *"gates"* and *"events"*. Gates allow or inhibit the passage of fault logic up the tree and show the relationship between events required for the occurrence of the higher level event. The event symbols used are shown in Figure 1.

Event Symbols **Name and Description**



Top event or intermediate event description box. These events are further developed by a logic gate

Basic event. This event is at the limit of resolution of the fault tree and does not require further development.

Conditional event. Specific restrictions or conditions that apply to a logic gate. Used primarily with PRIORITY AND and INHIBIT gates.

Undeveloped event. An event which is not further developed either because it is of insufficient consequence or because information is unavailable.

External or "House" event. An event that is expected to occur, or not occur, with certainty during operation. These events are not , of themselves, faults.

Transfer symbol. Indicates that the logic is developed further at another point.

Figure 1 Event Symbols Used in Fault Tree Analysis

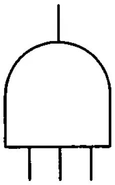
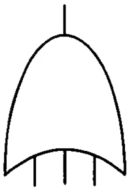
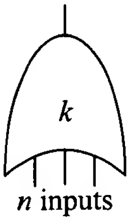
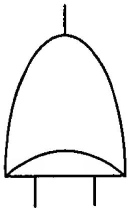
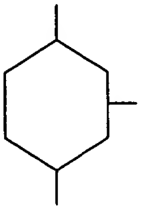
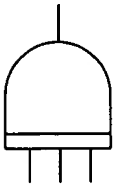
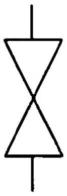
Gate Symbol	Name and Description
	**AND gate.** Output event occurs if all input events occur simultaneously.
	**OR gate.** Output event occurs if at least one input event occurs.
	***K* out of *n* gate.** Output event occurs if at least *k* out of *n* input events occurs.
	**Exclusive OR gate.** Output event occurs if one, but not both, input events occurs.
	**Inhibit gate.** Input produces output when conditional event exists.
	**Priority AND gate.** Output event occurs if all input events occur in order from left to right.
	**NOT gate.** Output event occurs if input event does not occur.

Figure 2 Gate Symbols Used in Fault Tree Analysis

The gate symbols used in fault tree construction are shown in Figure 2. The AND gate and the OR gate are the two most commonly used gates in fault tree analysis with other gates being specialized cases of the AND or the OR gate.

Fault Tree Construction

Fault tree construction begins with a clear definition of the top level event to be analyzed. Each fault tree considers only one of the many possible failure modes and therefore more than one fault tree may be constructed for a given system.

The next step is to clearly define the *external boundaries* of the system. The external boundaries are partially decided by the top level event being analyzed. An example often used is that of a telephone [2] [3]. If the top level event is that the bell is not loud enough to hear in all parts of the house than the external boundary will be limited to the telephone. If, on the other hand, the top level event involves noise on the line, the system boundary may include the wiring in the house and perhaps the local exchange.

The *limit of resolution* for the analysis must also be established. This is the point beyond which events will not be further developed. If, for instance, a water supply system were being analyzed, the failure of a pump might be the limit of resolution provided that adequate reliability data was supplied for the pump. The causes of the pump failure may not be important in the overall system analysis. The limit of resolution may also be established from considerations of feasibility. There is usually a finite limit to the time and cost afforded the analysis.

Once the system has been defined and the top level event has been selected, the fault tree is developed by determining the immediate, necessary, and sufficient causes for its occurrence. These are not the component level faults, but immediate cause of the event.

The immediate causes of the top level event are then treated as intermediate events and their immediate, necessary, and sufficient causes are determined. The tree is continually developed in this manner until the limit of resolution is reached.

While no set of rules can guarantee the construction of the correct fault tree in every circumstance, References [2] and [3] provide lists of rules which help ensure the fault tree is developed in a methodical manner. They are as follows:

1. Write the statements that are entered in the event boxes as faults; state precisely what the fault is and when it occurs.
2. If the fault in the event box can be caused by a component failure, classify the event as a "state-of-component" fault. If the fault cannot be caused by a component failure, classify the event as a "state-of-system" fault.
3. If the normal functioning of a component propagates a fault sequence, then it is assumed that the component functions normally. Do not assume that the miraculous failure of some component will block the propagation of a particular fault sequence.
4. All inputs to a particular gate should be completely defined before further development of any of them is undertaken.
5. Gate inputs should be properly defined fault events, described using rectangular boxes. Gates should not be directly connected to other gates.

To illustrate the construction of a simple fault tree, let us consider a the system shown in Figure 3. The system consists of a light used to illuminate a road sign. The light is usually powered by the city power supply but in the event of a power failure, the switch transfers over to the backup battery. The top event is "No Light Provided". The external boundary of the system will be limited to the light, wiring, switch, and battery. Failures of these items will be considered basic events, as will failure of the city power supply, and the tree will not be developed beyond this point.

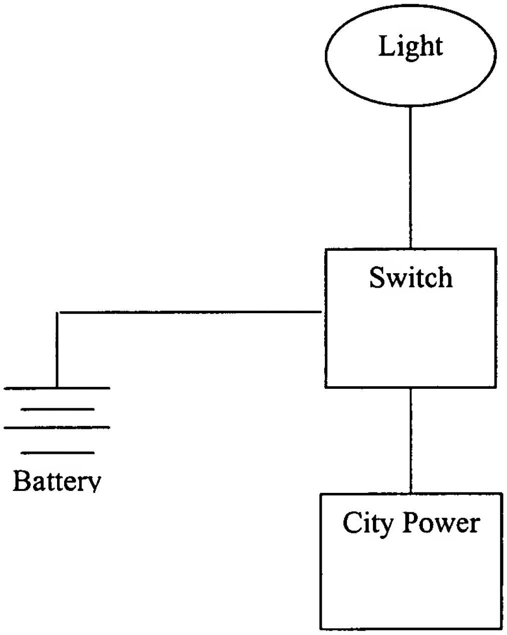


Figure 3 Road Sign Light

The immediate causes of the top event are:

"No Power" **OR** "Wiring Failure" **OR** "Bulb Failure"

Since any one of these faults occurring results in the top level event, they are grouped with an **OR** gate as shown in Figure 4.

"Wiring Failure" and "Bulb Failure" are basic events and are not further developed. The immediate causes of the "No Power" event are:

"City Power Failure" **AND** "Backup Power Failure"

Since both of these events must occur for the "No Power" event to occur, they are grouped by an **AND** gate.

"City Power Failure" is a basic event and is not further developed. The immediate causes of "Backup Power Failure" are:

"Battery Failure" **OR** "Switch Failure"

Since either of these events result in the occurrence of "Backup Power Failure" they are grouped by an **OR** gate. At this point, the tree has been developed to the limit of resolution. The completed tree is shown in Figure 4.

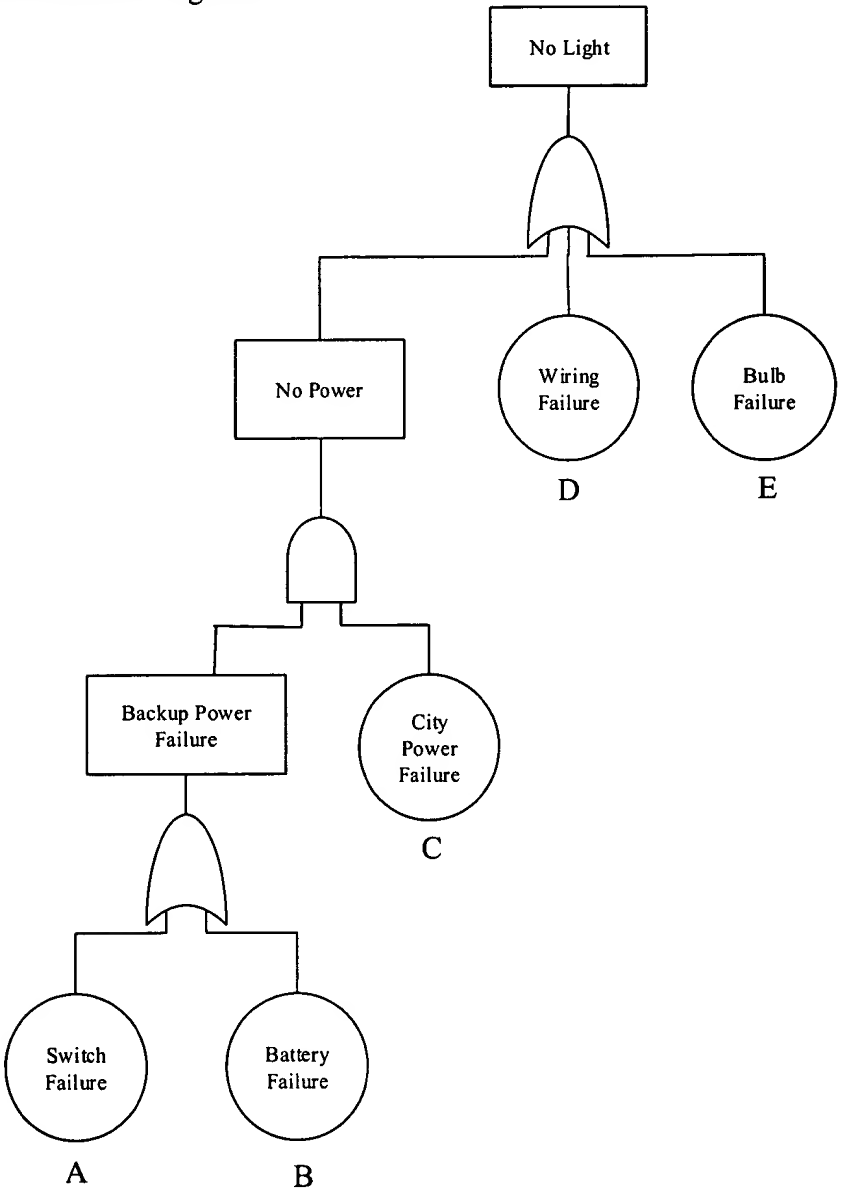


Figure 4 Fault Tree of Road Sign Light System

Quantitative Fault Tree Analysis

A quantitative analysis can be performed on a completed fault tree given sufficient data on the probabilities of basic events. If the fault tree for top event *T* contains independent basic events which appear only once in the tree structure, then the top event probability can be obtained by working the basic event probabilities up through the tree [2]. Intermediate event probabilities are calculated starting at the base of the tree and working upwards toward the to event. For events grouped by an AND logic gate, the higher level event probability is found from the equation [1]:

$$P_T = \prod_{i=1}^{n} P_i$$

For events grouped by an OR logic gate, the higher level event probability is found from the equation [1]:

$$P_T = 1 - \prod_{i=1}^{n} (1 - P_i)$$

To illustrate the use of these equations, we can again use the simple fault tree shown in Figure 4. Suppose the probability of battery failure is 0.002 and the probability of switch failure is 0.001. Then the probability of backup power failure is:

$$P = 1 - (1 - 0.002)(1 - 0.001) = 0.002998$$

If the probability of city power failure is 0.001 then the probability of no power is:

$$P = (0.002998)(0.001) = 2.998\text{E-}6$$

If the probability of wiring failure and bulb failure are 0.001 and 0.02 respectively then the probability of the top event is:

$$P = 1 - (1 - 2.998\text{E-}6)(1 - 0.001)(1\text{-}0.02) = 0.02098$$

Unfortunately, the approach shown above can only be applied to a small category of fault trees which do not contain repeated events. When a fault tree has repeated events, gate events no longer occur independently and this method will not produce accurate results.

When fault trees have basic events that appear more than once, the *minimal cut sets* method is most often used to obtain the top event probability [2]. Each cut in the minimal cuts set is a combination of events that result in the top event. As an example, the minimum cuts set for the Figure 4 fault tree would be AC, BC, D, and E. Each included set must not be a subset of another set. For instance BCE would not be included since E alone results in the top event. If a fault tree has n minimal cut sets K_i, $i = 1,2,\ldots,n$ then the top event exists if at least one minimal cut set exists.

$$P(T) = P\left(\bigcup_{i=1}^{n} K_i\right)$$

The top event probability can then be calculated using the *"Inclusion-Exclusion Expansion"[2]*

$$P(E_1 \cup E_2 \cup \ldots \cup E_n) = \sum_{i=1}^{n} P(E_i) - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} P(E_i \cap E_j) + \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=j+1}^{n} P(E_i \cap E_j \cap E_k) + \ldots + (-1)^{n+1} P(E_1 \cap E_2 \cap \ldots \cap E_n)$$

This equation can be extended to any number of events. To illustrate the use of the equation, we will once again return to the fault tree shown in Figure 4 which has minimal cut sets:

AC, BC, D, E

The basic event probabilities were previously given as :

$$P(A) = 0.001 \quad P(B) = 0.002 \quad P(C) = 0.001 \quad P(D) = 0.001 \; P(E) = 0.02$$

In this example, the expansion will have four terms. Applying the expansion to the cut sets gives:

$$P(T) = [P(AC) + P(BC) + P(D) + P(E)]$$

$$- [P(ABC) + P(ACD) + P(ACE) + P(BCD) + P(BCE) + P(DE)]$$

$$+[P(ABCD) + P(ABCE) + P(BCDE)] - [P(ABCDE)]$$

$$P(T) = 0.021003 - 0.000020065 + 8.2E\text{-}11 - 4.0E\text{-}14$$

$$P(T) = 0.0209829$$

This is the same result obtained previously. This example shows that calculating each term in the inclusion-exclusion expansion can be tedious and time consuming. This example dealt with only four cut sets where complex fault trees may deal with tens of thousands of cut sets. Fortunately, the failure probabilities are typically small numbers so the product of many such numbers begins to become insignificant in the final answer. This causes the first term to be more significant than the second term which is more significant that the third term and so on. The convergence of the expansion is shown in Figure 5.

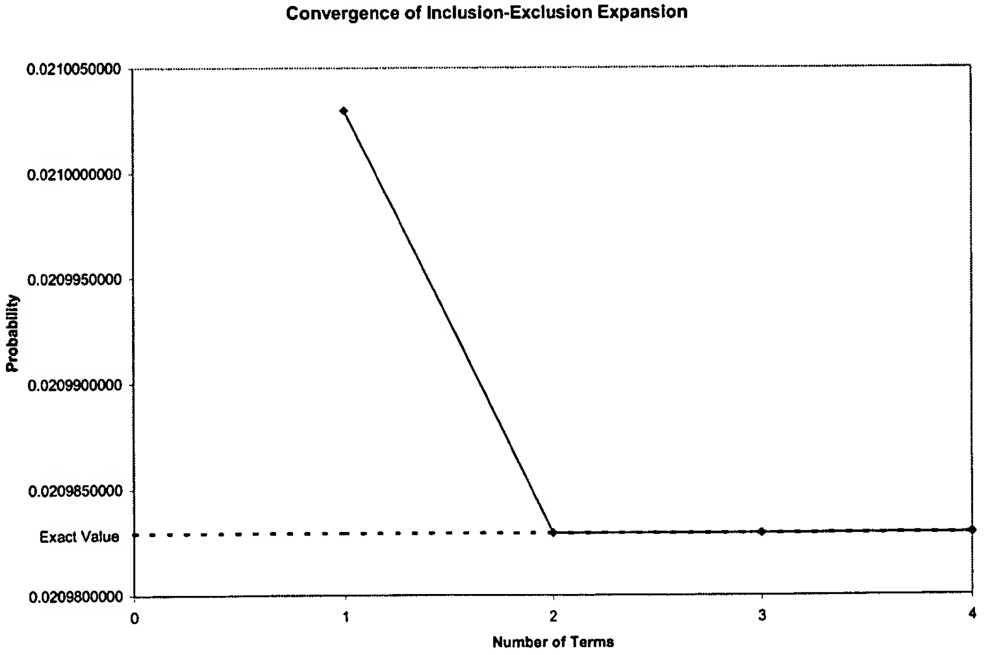


Figure 5

The series evaluation adds odd numbered terms and subtracts even-numbered terms, each successive term being numerically less significant. Therefore, truncating the series at an odd-numbered term will provide an upper bound and truncating the series at an even numbered term will provide a lower bound for the probability.

Summary

- Fault tree analysis is a very useful analytical tool for both qualitative and quantitative analysis.
- Fault tree analysis is a top down approach that evaluates one failure mode at a time and is useful for determining the root cause of the failure.
- The graphical nature of a fault tree makes it an excellent way of presenting data as well as flow chart for performing reliability calculations.
- Fault tree analysis requires a thorough understanding of the system and for large systems, will most likely require input from multiple disciplines.
- Although reliability calculations based on fault trees can become quite cumbersome, approximate methods which yield accurate results are available.
- Before constructing a fault tree, is very important to clearly describe the top event, the system boundaries, and the limits of resolution.
- Fault tree analysis should be performed as early in the design cycle as possible. This will allow changes to the design based on results of the analysis.

- **References**

1. K. C. Kapur, INDE 526 Reliability in Product Design & Testing Course Lecture Notes, University of Washington, Seattle, WA, 2001, Module 8

2. J. D. Andrews & T. R. Moss, Reliability and Risk Assessment, AMSE Press (2nd edition), 2002 (ISBN 0-7918-0183-7) pp 201-268

3. N. H. Roberts, W. E. Vesely, D. F. Haasl & F. F. Goldberg, Fault Tree Handbook, NUREG-0492, U.S. Nuclear Regulatory Commission, 1981

4. C. A. Ericson II, Course No. 9Sv276 Fault Tree Analysis Student Workbook, D&SG Employee Training and Development, The Boeing Company, 1999

5. S. Contini, "A New Hybrid Method For Fault Tree Analysis", Reliability Engineering and System Safety, **49** (1995) 13-21

6. K. A. Reay and J. D. Andrews, "A Fault Tree Analysis Strategy Using Binary Decision Diagrams", Reliability Engineering and System Safety, **78** (2002) 45-56

7. R. M. Sinnamon and J. D. Andrews, "New Approaches to Evaluating Fault Trees", Reliability Engineering and System Safety, **58** (1997) 89-96

8. A. E. Summers, "Viewpoint on ISA TR84.0.02 – Simplified Methods and Fault Tree Analysis", ISA Transactions, **39** (2000) 125-131

9. I. D. Walker and J. R. Cavallaro, "Failure Mode Analysis for a Hazardous Waste Clean-Up Manipulator", Reliability Engineering and System Safety, **53** (1996) 277-290

Appendix D

Relex Software

MANAGING RELIABILITY ACROSS THE PRODUCT LIFECYCLE™

ISO 9001 CERTIFIED TickIT 2000

About Us | Products | Services | Customers | News & Events | Customer Support | Resources | Send Page ✉

Printer Frier

- **Relex Resources**
- **Reliability Overview**
  - Reliability Engineering
  - Reliability Analysis
    - - Reliability Predictio>
    - - Reliability Prediction Model>
    - - Event Tre>
    - - Fault Tre>
    - - FMEA/FMEC>
    - - FRACA>
    - - Life Cycle Cos>
    - - Maintainability Predictio>
    - - Marko>
    - - Reliability Block Diagra>
    - - System Optimizatio> and Simulation
    - - Weibul>
  - Glossary of Terms
- **Reliability Documents**
- **Additional Sources**
- **Reliability Discussion Forum**

Home > Resources > Reliability Overview > Reliability Analysis > Fault Tree

What is Fault Tree Analysis?

A fault tree analysis (FTA) is a deductive, top-down method of analyzing system desig and performance. It involves specifying a top event to analyze (such as a fire), followe by identifying all of the associated elements in the system that could cause that top event to occur.

Fault trees provide a convenient symbolic representation of the combination of events resulting in the occurrence of the top event. Events and gates in fault tree analysis are represented by symbols.

Fault tree analyses are generally performed graphically using a logical structure of AN and OR gates. Sometimes certain elements, or basic events, may need to occur together in order for that top event to occur. In this case, these events would be arranged under an AND gate, meaning that all of the basic events would need to occu to trigger the top event. If the basic events alone would trigger the top event, then the would be grouped under an OR gate. The entire system as well as human interactions would be analyzed when performing a fault tree analysis.

Relex Fault Tree/Event Tree

Relex Fault Tree/Event Tree enables you to quickly and conveniently create complete fault trees. You can enter gates and events and assign their individual properties usin just a few mouse clicks. When finished, the software performs fast and accurate calculations analyzing your system safety - computing the unavailability of the top eve and determining which events are most likely to cause the top undesirable event.

Contact Relex

Phone: 724.836.8800 | Request Info | Email: info@relexsoftware.com

Site Map | Search | Feedback | Privacy

Appendix E

Safety Critical Systems Analysis

Carnegie Mellon University
18-849b Dependable Embedded Systems
Spring 1998
Authors: Robert Slater

Abstract:

Safety analysis is a method for evaluating the hazards and risks posed by a system and ways to minimize them. Many guidelines exist to guide safety analyses, but all study two main areas. Hazard analysis is the first stage, in which the system is studied for situations in which potential harm could result, and the frequency with which those situations occur. Risk analysis is the second stage, in which the possible outcomes of the hazards and the frequency of appearance of each outcome is determined. This allows sources of potential harm in the system to be prioritized and dealt with to increase the safety of the system. Many standards exist for acceptable levels of safety in different industries, but sometimes it is a judgement call as to when the system is safe enough. In many cases, the best safety analyses are performed by those expert in the analysis techniques, and novices are best tutored in the techniques before performing them independently. For embedded systems in which there is the potential for harm to a person or the environment safety analysis can be a useful way to quantify that potential and minimize it, but its most effective use lies in the hands of those familiar with it.

Contents:

Introduction

Safety critical systems exist all around us, from nuclear power plants to chemical processing plants to heart monitors and emergency phone systems such as 911 in the United States. The industries that support these systems have put a great deal of deliberation and thought into making their systems as safe as possible, both in providing their designed function and in preventing their malfunctioning or defects. What remains a difficult task, however, is quantifying the safety inherent in a system. Safety is a nebulous concept, and is therefore difficult to define or measure. Should safety be measured in the amount of harm done, or perhaps a ration of the amount of harm done vs. the potential to do harm? Over what time span does one measure the harm a system can do? And how can one system be said to be safer than another? While these industries have no universal answer to these questions, they do have a collection of techniques which helps address them. Known as safety-critical systems analyses, these techniques can be used to assess the level of safety inherent in a system, and possible improvements that can be made. Furthermore, they take steps towards addressing those questions of measurability and quantification that seem so intractable.

This analysis typically takes two forms, hazard analysis and risk analysis. Hazard analysis is the examination of a system for potential to cause harm. In it the system or a model of the system is examined for ways in which it can cause harm or dangerous situations. Risk analysis examines the potential damage that can result from the hazards present in a system.[Storey] It examines the types of harm that can occur in hazardous situations caused by the system and their likelihood of occurrence. When tied together, the two forms of analysis can provide a detailed and potentially prioritized list of the potential harm that a system can cause. That list can be used in an iterative design process to refine the system and add safeguards for particularly dangerous outcomes. It is the combination of these two forms of analysis that proves most effective, and it is their use throughout the design process that produces the safest systems.

For embedded systems designers, safety analysis is relevant because of the increasing automation of safety critical systems. The increases in adaptability and response time are too attractive not to include. However, as we are increasingly finding in systems in the field, the added complexity of automation causes unsafe conditions to be overlooked or improperly protected against. As projects like the drive-by-wire car and automated emergency-call systems become feasible, embedded system designers will need the methodology of safety-critical analysis to ensure that their systems meet the safety requirements for these kinds of applications. They introduce guidelines, quantification, and methodical process to an unclear aspect of design that embedded designers may not initially be equipped to handle on their own.

Key Concepts

As stated above, there are two main branches safety critical system analysis, hazard analysis and risk analysis. We will discuss both of these, in addition to the fuzzy human factors that make the area difficult to approach.

Hazard Analysis

A hazard is a situation in which there is actual or potential danger to people or the environment.[Storey] Hazard analysis, accordingly is a method for examining a system to examine how it can cause hazards to occur, and, in some cases, how to prevent those hazards from occurring. While the actual techniques may vary in their approaches, they all have certain aspects in common. They typically will have a suggested model of the system to use, which hopefully exposes the activity and components of the system in a meaningful way so as to examine them for hazards. They will have a method of examining the different parts of the model that is systematic and attempts to be as complete as possible. This methodology will typically have a formatted sort of result for easy interpretation, so that the results can be read without going through the preparer's thinking. Lastly, the analysis technique may have additional guidelines for the process of making the analysis.

The system model is one of the most crucial parts of the analysis, second only, perhaps, to the actual method of examination. It places limits on the way the system is examined, and those limits can be fatal weaknesses at the heart of an analysis. Limits, in the forms of levels of abstraction, are necessary to allow people to perform the analysis, as humans are bad at dealing with high levels of complexity. The danger exists, however, that potential hazards will be hidden in the abstractions, and thereby go unexamined. It is important, therefore, that the model for the system extend to the appropriate levels of detail as well as supporting analysis at higher levels of abstraction.

The next portion of hazard analysis is the actual mechanism of analysis. Typically this will involve taking apart the system model and examining each portion and interaction of the model for hazards that might be caused by that component. This methodology should address every portion of the system, in addition to ensuring that each section is examined adequately. Some methodologies will go so far as to have checklists or forms to fill out for each component or hazard. Once the examination has been performed, these forms or diagrams or other tool of the method can be used to quickly summarize the result. This organized visual feedback can be the most powerful part of the analysis, making the results of the study available and understandable with a much lower investment of time and effort.

Finally, the more developed and involved methods of analysis will have guidelines of a greater scope, concerning the process rather than the procedure of the analysis. In many cases a team is specified to perform the analysis, preferably with multiple areas of expertise so as to cover all aspects of the system. The amount of time spent in meetings and frequency of meetings is even specified in some types of analysis. The important things that these guidelines provide are ensuring expertise in the proper areas and diligence in performing the analysis. A team can double check the results of one member, and discuss points, and preventing overwork or exhaustion ensures that the team can apply the proper concentration to the analysis. Also, there is typically an admonition to employ the analysis throughout the design cycle. Early use of analysis can prevent the large hazards, and continued use will further refine and improve the system.

There also exists a branch of hazard analysis called probabilistic hazard analysis, which attempts to place a chance of occurrence on each hazard in addition to identifying it. This can be based on field data, component lifetimes, standards, or any other numerical data which give some idea of the conditions the system is likely to be placed in, and the behavior of the system in those conditions. Structural and mechanical analyses tend to use load and capacity distributions to calculate probabilities of failure, but a large number of statistical methods exist for this kind of analysis.[Blockley] What is important to know is the statistical reasoning behind the method used, and that the end result is the capacity to know to some degree of accuracy the probability of a hazard actually occurring with the system in operation.

Risk Analysis

Risk is a combination of the frequency or probability of a specified event, and its consequence. Risk analysis is the counterpart to hazard analysis, taking a list of hazards and producing a list of possible outcomes and their likelihood of happening. Classically, probabilistic risk analysis is used to describe this process, while risk analysis refers only to the examination of outcomes. In common usage, however, risk analysis without probabilities is hardly ever performed, and so risk analysis is typically used to refer to the combination of the two.[Blockley]

The first part of risk analysis is an examination of the possible results of a hazard. Many hazards can produce a range of actual harm done, and so each hazard must be examined to determine the possibilities. Often these can be categorized into different levels of harm, distinguished by the amount of harm done.[Storey] Multiple fatalities would be more serious than a single fatality, a fatality more serious than a major injury, and so on. In this way the analysis can be speeded up by needing to be less precise without losing the major insight as to how much harm can be done.

Once this enumeration of possibilities has been done, the analysis can proceed into its probabilistic phase. Each potential harm is associated with a probability of occurring. Much like probabilistic hazard analysis, the numerical data can be derived from field data, component lifetimes, and other sources, and manipulated statistically. These calculation tend not to be absolutely correct, but the relative probability of two events is typically preserved. Thus, if an injury occurs twice a year and a fatality once every five years according to the calculations, while those might not be the actual observed rates of occurrence, injuries should tend to occur more often than fatalities. A similar categorization can be performed as above, in which ranges of probability are lumped together. Again, the intuition as to frequency of occurrence is preserved, even if the absolute numbers are not. Thus each hazard has a list of possible harms done and probabilities of each.

The combination of these two values can be an extremely valuable tool for prioritizing further work and determining when the system is safe enough. Highly damaging and likely hazards can be addressed in the refinement of the design before rarer or less dangerous ones. Not only are resources used more efficiently to improve safety in this way, but they are also target at the problems most likely to cause the system to be dangerous. It can also be used to establish the break-even point, when the cost of developing further safety is greater than the cost of dealing with the harm caused by the hazard. Once this line is crossed where the system is 'safe enough', further safety work can be held off until the next design cycle.

Fuzzy Societal Factors

There are shortcomings, however, to safety analysis, and the greatest are the poor definition of the problem to be solved, the heavy dependence upon the expertise of the annalists, and the indeterminate nature of the results of the investigation. Safety is not an easily measurable or definable property to a system, and therefore it is difficult to separate the safety properties or components of a system from the normal functioning ones. Since in many cases safety is an emergent property of the system as a whole, and dependant upon its interaction with its environment, which may be changing or ill-defined, the task of isolating and describing the 'safety' of the system becomes incredibly difficult. Those performing the analysis must have insight into the operation of the system and what it means to be safe, and the quality of the analysis is directly dependant upon the quality of this insight. If the annalists are inexperienced, or not familiar with the ways in which systems present risks, then the analysis will suffer, and no methodology can completely make up for shortcomings of those performing the analysis. Finally, once the results have been gathered, because of the poorly defined nature of safety, it is difficult to interpret the results or come to any conclusions regarding their correctness. All too often the true test of how safe a system is occurs in the field, and the best measure we have is the hazards and harm it does cause, rather than the ones the analysis has prevented.[Leveson]

It is difficult to answer the question of what safety is, or how it applies to a specific system. One definition is that it is a property of a system that it will not endanger human life or the environment.[Storey] Defining danger to human life or the environment, though, is an exercise in enumeration, rather than a procession from basic principles, and drawing the line between what is dangerous and non-dangerous is often a matter debate. Is one particle per million of a certain chemical dangerous or not? Is 55 m.p.h. the best speed limit, or does the improved performance of modern cars warrant an increase in the speed limit? What is under debate here is the tradeoff between safety and other concerns, such as cost or performance. It is easy to measure the cost of better containment of chemicals, or the time saved in having a higher speed limit. Higher cancer rates in an area, or a higher number of highway accidents and fatalities, however, are less easy to measure, although still quantifiable, but how are the two compared?

The answer is: on an ad hoc basis. Those performing the analysis must consider each part of the system and each interaction, and relate it to its environment, and then consider how it might cause damage. If there is no one with experience regarding that particular component or subsystem, then the analysis of that particular component has a large possibility of being inadequate. Furthermore, when the next level of abstraction is reached, more hazards my be hidden behind that abstraction, as the interactions of the component have larger effects that are not accounted for. Expertise in that particular component's operation is important to understand its larger purpose in the system. Expertise in its interactions with the environment is essentially expertise in the safety aspects of that component. If either is lacking, then the analysis will be incomplete. Unfortunately, while many people may be expert in the operation of a component or sub-system, knowledge about interactions with people and the environment is not necessarily as widespread. Thus we see in many types of safety analysis the mention that experienced practitioners provide the best analyses, and training is necessary to allow others to practice safety analysis effectively on their own. This safety expertise is still held by people, rather than having been encapsulated within the analytical methods themselves, and will remain so for the foreseeable future.[Leveson]

Given competent practitioners, however, and a substantial looking analysis, how does one view the results? It is difficult to accept any guarantees of completeness, as even a simple system interacting with a typical environment has many complicated interactions, many of which are poorly understood and potentially dangerous. It is often small details that are overlooked that combine to cause danger and, in some cases, disaster. Often the basic assumptions behind a particular part of the analysis may be flawed, such as the performance required of a bearing when in the field the system will be used beyond the designed parameters, putting undue stress on the bearing. If the system is operated outside the specified environment, then the analysis is invalid. Overlooking an interaction or making a mistaken assumption is too easy to not occur at least once in an analysis.

Quantization of results is also a problem. While the analysis may provide numbers that help define how dangerous a risk is, those numbers may have no bearing in reality. They may be based upon older systems that have some fundamental or perhaps subtle difference in their operation. Often, if no data exists, the numbers may be based on the gut instinct of the person performing the analysis. Such a stab in the dark may be fairly accurate if made by someone competent, but even competent people and analysts can make mistakes, or misjudge dangers based on personal prejudices. Human error and the difficulty of prediction make this stage of the analysis difficult to have great confidence in.

And even if the analysis is complete, and the risks defined and prioritized, it is also difficult to determine what safety level is great enough. Sometimes regulating agencies will have guidelines or standards to follow, which make the decision easy. Other times, however, the tradeoffs are more difficult to determine, and the decisions harder to make. When is the cost of making a car safer greater than the cost of the lawsuits it prevents? And if there is a societal cost in terms of medical expenses, work lost, emotional trauma, and recovery work, who is accounting for that? Often this falls upon a project manager or an executive who is not expert in the system. Sometimes the guarantees that must be met for increased safety impact time to market, or the performance needed from a piece of equipment, rather than mere dollar cost of the system. The other side of the equation is usually well defined, and the benefits of increased safety difficult to conceptualize.

What all of this amounts to is that the problem of safety is ill defined, and our current attempts to address it are incomplete. Those who have addressed or concerned themselves with safety in the past will be able to use that experience to aid further work, but those without that experience will not have enough insight into safety concerns to do as effective a job. The results are unverifiable, though in general some level of confidence will be able to be held in them. Finally, the tradeoffs between safety and other concerns such as cost and performance are sometimes difficult to perceive, and complete safety is almost always unattainable.

Available tools, techniques, and metrics

There are many forms of safety analysis, and some of the major ones are discussed below. Some are useful tools or techniques, while others are designed to be comprehensive. All of the analytical methods listed below have been used to good effect in the past. If one of them seems applicable to a particular problem or system, seek out further sources and, preferably, someone who has used that method in the past for further guidance.

Checklists

Despite their seeming simplicity, checklists are a form of safety analysis. As an example, an airplane is a safety critical system. As one level of analysis, a pilot must complete a pre-flight checklist before flight to ensure that the plane is working properly. This checklist is a simple form of safety analysis. They are generally useful where a problem is well understood, and examination rather than system analysis is the goal.

Fault Tree Analysis

Fault trees developed in the aerospace industries, but have found uses in many areas, most recently software analysis. Fault trees operate by developing a list of the faults that can occur in a system, and attempting to trace them back to their root causes. The reason that they are called fault trees is that there is a tree-like formal notation that accompanies the analysis, in which different types of events are specified by differently shaped containers, and the events are linked logically in tree like structures to lead up to the eventual fault of the system. While this method can be used to show complicated interactions, it is still subject to the danger of overlooking aspects of the system as these are mostly enumerated. It is advisable to combine this with another more methodical approach to ensure the completeness of the analysis.. An example is shown below.[Leveson]

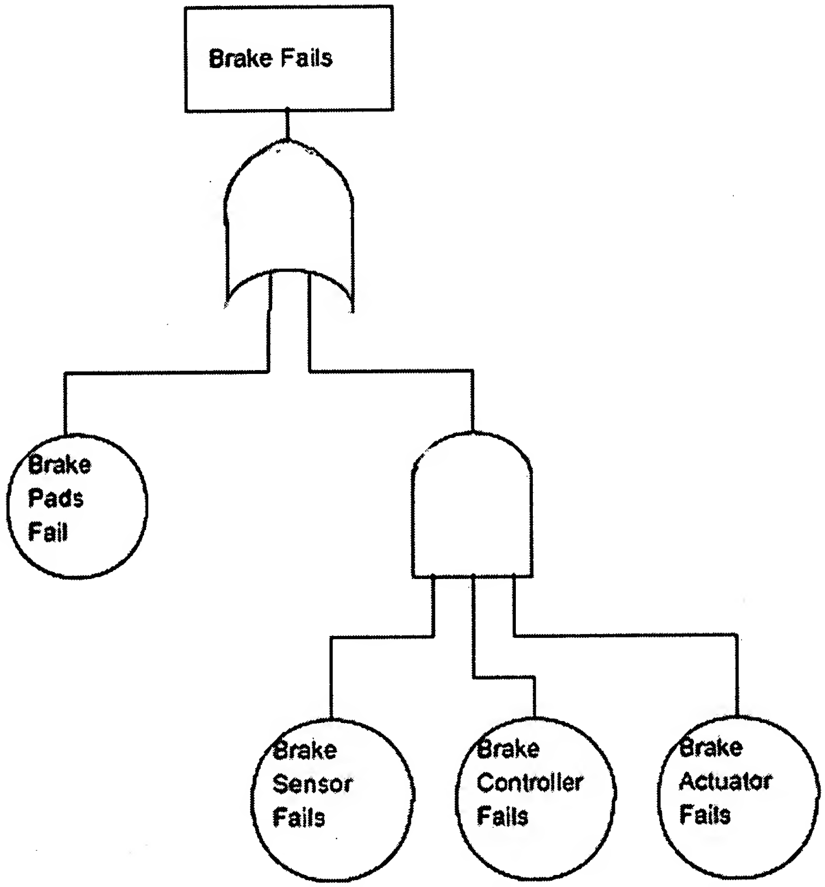


Figure 2: Example of a simple fault tree for a brake system

Event Tree Analysis

Event trees function similarly to fault trees, but in the opposite direction. An event tree attempts to enumerate a list of components and subsystems and determine the result of their operation or non-operation. In this way all sequences of possible events are covered involving those components. As with fault trees, enumeration is the main form of choosing subsystems and components to examine, so a more methodical approach should be coupled with event tree analysis for greater completeness. An example is shown below.[Storey]

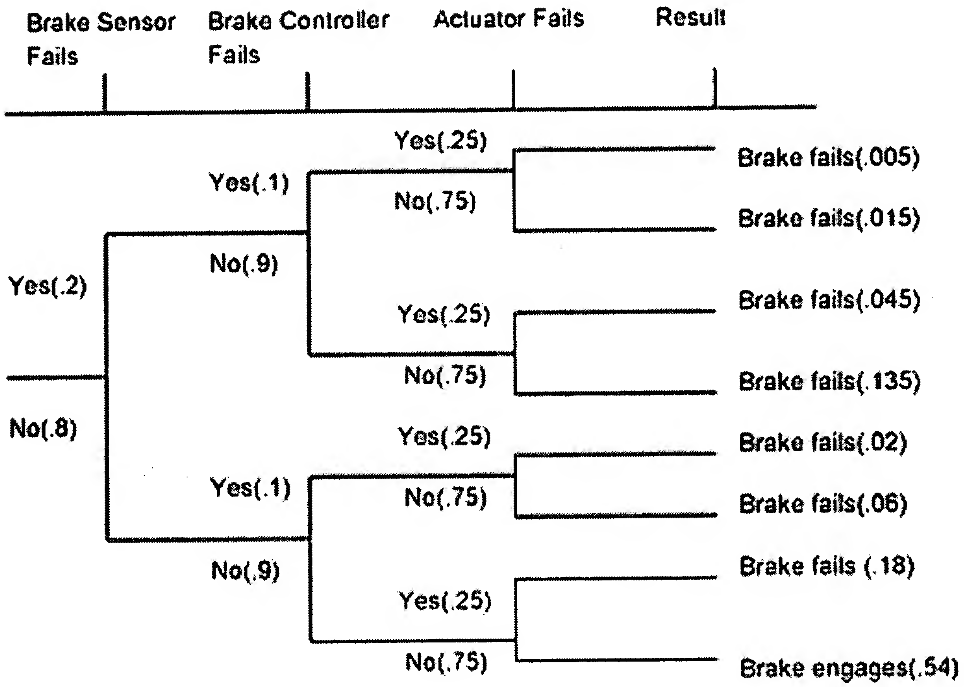


Figure 1: Example of an event tree for a brake system

Failure Modes and Effects (and Criticality) Analysis

Failure Modes and Effects Analysis (FMEA) and Failure Modes and Effects and Criticality Analysis (FMECA) function much like a checklist, only a more organized one. There is a standard form which must be filled out, in which each subsystem or component is listed, along with the different ways in which that particular component can fail. Once these failure modes have been listed, the effects of that failure are listed. In the criticality analysis, each failure mode is associated with a frequency, and each effect with a 'danger rating'. These numbers are used to provide some idea of exactly how much risk that failure mode places upon users or the environment. Once these have been collected, each failure mode has a possible protective measure listed with it. Criticality analysis adds a cost of protection number here. This provides a list of hazards, risks, and possible countermeasures, and the criticality analysis orders them according to the level of danger they represent. The danger here is, again, that of leaving something out in the course of listing the possibilities.[Storey]

HAZard OPerability Studies

HAZard Operability Studies (HAZOPS) is a methodology for safety analysis that is highly rigorous, precise, and involved. A system model is constructed and each component is described with a list of attributes that describe the operation of that component. A list of guidewords with well defined meanings is then applied to each attribute to determine the effect of the deviation from normal operating described by that attribute. For example a pipe might have the attribute flow, for which the guideword backwards would mean backwards flow through the pipe. By having a well defined set of guidewords and a good system model, as well as expert annalists, this method attempts to be completely rigorous in its application. In addition to this mechanism of analysis, it also has guidelines describing the process of meeting and conducting the analysis.[MoD]

Relationship to other topics

Ultra-dependability

Since ultra-dependable systems cannot be tested to assure the requisite level of reliability, safety analysis could prove useful as a mean of determining the safety level of such a system. The only problem is the amount of trust which can be placed in the analysis.

Software Reliability, Software Fault Tolerance, Software Safety

Safety analysis is being targeted all of these aspects of software in order to improve its quality and address concerns that, up until now, have not really been applied to software.

Multi-disciplinary Design

Since most systems being analyzed for safety purposes will have subsystems designed by different specialties or professions, from electrical, chemical, and mechanical engineers to medical professionals, multi-disciplinary knowledge is necessary to construct a good system model and to examine that model.

Social & Legal Concerns, Ethics

The question of how safe is safe enough is often answered in the social or legal area. In come areas, ethics may delineate risks which are monetarily tenable but still need to be protected against.

Validation, Verification, & Certification

In the process of certifying systems for safety, safety analysis is often used as a means of constructing a safety case, or persuasive argument that the system is safe.

Conclusions

Safety Critical Systems Analysis is an attempt to solve a poorly defined problem. Safety is an ill-defined property of a system, and one that can rarely be confined to one portion of the system. Therefore considering the safety of a system involves examining the system as a whole, and its interactions, a task to which people are ill suited. Instead, these analyses break the system down into manageable parts and break the analysis into easily manageable parts. The take the model and its components, and examine each for the hazards they represent. Then the hazards are examined for the potential risks they present. Together they represent a list of the danger the system presents to people and its environment, and also prioritize those dangers in their severity and in the necessity of their being prevented.

What it is necessary to remember is that the analysis is being conducted by humans, who are prone to error. Components or interactions can be poorly understood. Hazards can be overlooked. Risks can be over- or underestimated. And the results, finally, are subject to human judgement as to how safe is safe enough, and how much work should be put into making the system safer. Those with experience in safety analysis will do a better job of analysis, and those with a greater interest in safety will err on the side of caution. The best thing that can be said about safety critical systems analysis is that, despite its imperfections, it has a history of being successful, and in some cases highly so. It is our best tool in addressing a problem that is poorly understood, and far too often undervalued.

Annotated Reference List

- Blockley, David. "Engineering Safety." Mcgraw Hill, 1992.

This is a very complete book detailing a lot of the basic ideas in safety analysis from the point of view of civil engineering. While it doesn't address embedded concerns directly, many of the idea are applicable.

Leveson, Nancy. "Safeware: System Safety and Computers." Addison Wesley, 1995.

This book covers the area of safety analysis from the perspective of computer systems. It begins to address safety analysis to the problems faced by computers and software, but these methods have not yet been proven in this field. It also covers a lot of other issues related to safety analysis, hazard, and risks, and is probably a good text for the computer professional who has to deal with safety issues, but does not want to delve too deeply into the material.

- MoD "Interim Defense Standard 00-58 HAZOP Studies on Systems Containint Programmable Electronics." Aug. 1996

This standard details the application of the HAZOPS approach for electrical and computing systems. It provides more detail into the HAZOPS procedure and potential ways of adapting it for embedded applications. A copy of this, and other safety-related standards can be found at http:://www.seasys.demon.co.uk/

- Storey, Neil. "Safety-Critical Computer Systems" Addison Wesley, 1996.

This book covers the basics of safety and safety analysis in its initial chapters, but provides more of a surface skim than an in depth look.

Go To Project Page

Appendix F

DRAWINGS SHOWING PROPOSED CHANGES

IN MARK-UP FORM USING

RED INK

None.